UNIVERSITY OF ALBERTA

Dynamic Mathematical Morphology and Its Applications

By

© Jun Yang

A thesis
submitted to the Faculty of Graduate Studies and Research
in partial fulfillment of the requirements for the degree
of Master of Science

Department of Computing Science

Edmonton, Alberta
Spring 1995

ISBN 0-612-01665-X

UNIVERSITY OF ALBERTA

RELEASE FORM

NAME OF AUTHOR: Jun Yang

TITLE OF THESIS: Dynamic Mathematical Morphology and Its Applications

DEGREE: Master of Science

YEAR THIS DEGREE GRANTED: 1995

(Signed) ____________________

Permanent Address:
#203 10317, 107 AVE
Edmonton T5H 0V7, Alberta
Canada

Date: April 21, 1995

UNIVERSITY OF ALBERTA

FACULTY OF GRADUATE STUDIES AND RESEARCH

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research for acceptance, a thesis entitled **Dynamic Mathematical Morphology and Its Applications** submitted by **Jun Yang** in partial fulfillment of the requirements for the degree of Master of Science.

Xiaobo Li

Supervisor: Dr. X. Li

Examiner: Dr. A. Basu

Jack

External: Dr. J. Mowchenko (Electrical Engineering)

Date: April 20, 95

To my parents,

and my wife

Abstract

Recently mathematical morphology has attracted more and more attention in the image processing field. Morphological operations transform an image with a structuring element which is a small pattern. They deal with shapes in the image directly and provide a representation of the image by conveying its geometrical information such as size, orientation and connectivity. This property makes mathematical morphology a natural approach to many imaging and vision applications.

In some cases regions of interest only occupy a small portion of an image, but traditional morphology applies operations to the entire image. This thesis proposes *dynamic mathematical morphology* which only operates on the pixels of interest and reacts to certain features of the image. Compared with traditional morphology, the computational load is reduced and the operation precision can be increased. In traditional morphological operations all pixels are processed one after another in a regular fashion which is usually the scanning order. Dynamic morphology analyses information within the image neighborhood designated by the structuring element and decides the next position dynamically. Its operation is performed along a specific path which cannot be decided before hand.

Dynamic morphology is designed for use in detecting objects from an image. The closed external boundary of an object contains essential shape information of the object and its detection is a key step in image understanding. Detecting moving objects with a static camera is an important problem of computer vision. Its applications include traffic control, surveillance, video phone and video conference. Based on dynamic morphology, new methods of continuous external boundary detection and moving object detection are proposed and implemented. The experimental results are encouraging.

Acknowledgements

First of all, I would like to express my deepest gratitude to my supervisor, Dr. Xiaobo Li, for his invaluable guidance, inspiration, encouragement and patience throughout the research work and the preparation of this thesis. His constant striving for creative and rigorous research influenced many aspects of my life. I benefited greatly from his profound knowledge and experience in the field. His insights and talents as well as his unfailing confidence in my motivation and capability made this work possible. I owe him the biggest debt of gratitude which cannot be expressed enough.

Thanks are due to Dr. Jack Mowchenko for his careful reading and valuable comments. Thanks also goto Paul Iglinski for his proofreading the abstract and the introduction of the thesis. I Thank Dr. Piotr Rudnicki for chairing my defense. I am grateful to all my friends in and outside the Computing Science Department for their help. I would also like to thank Dr. Xiaobo Li and the Computing Science Department for the financial support.

Finally I am deeply indebted to my parents, my parents in law, my wife, my sister and my brother. Without their immeasurable love, support and encouragement, I would not have achieved what I did. In particular, thanks to my wife, Jingrong Tang, for being with me through all the time and sharing all my feelings. Her love and understanding make my life rich and happy.

Contents

List of Figures

Chapter 1

Introduction

Mathematical morphology is a nonlinear image processing technique. It is composed of a series of operations based on set theory. In biology "morphology" refers to the study of the form and structure of animals and plants. Mathematical morphology provides a method for quantitatively describing and representing the geometrical structure of an image signal. Using mathematical morphology, the shape information in an image can be extracted by transforming it with another smaller and simpler image. This smaller image is usually called a structuring element. A structuring element can be considered as a probe or filter in the image being processed. Its shape and size are key factors in a morphological operation. In recent years morphological image processing has received more and more attention and has grown from a specialized discipline to a major research field.

In image processing, detection and analysis of the shape properties of an object often plays an important role. The object is identified by its features which correlate directly with shape. Morphological operations tend to simplify an image while keeping its important characters and eliminating the trivial ones. Mathematical morphology operates directly on shapes in an image. This makes it a natural processing approach and a valuable tool in machine vision. Morphological image analysis is employed in a variety of applications such as feature extraction, recognition, image coding and nonlinear filtering.

In traditional morphology, an operation is applied to all pixels of an image in a regular order. For some applications only specific parts or features of the image are of interest. It is a waste of computation sources to perform the operation on irrelevant pixels. The result of computations on the irrelevant region may affect the operation on the area of interest in the image and degrade the quality of the result. Doughterty [10] suggested a conditional dilation which limits the dilation into some regions when needed. A conditioning region is used as a restriction on the dilation result. This region should be known before the dilation begins. In practice however it is usually not possible to determine this region before hand.

To solve these problems a dynamic condition is introduced and incorporated into the traditional morphology. The condition specifies the properties of the pixels which will be processed. It is said to be *dynamic* because it uses the previous operation's result. This leads to a new concept in mathematical morphology. The modified morphology with its condition is called *dynamic morphology.* In a dynamic morphological operation, movement of the structuring element is directed by the output of applying the condition to each potential position. At each step of the operation, information of the image area covered by the structuring element is checked according to the condition. Next position of the structuring element origin is decided upon if the condition is satisfied. By a well designed condition, the morphological operation will only go through an adaptively selected path and process the pixels or regions that have some specific properties. This will reduce the computational cost and prevent the result from being affected by irrelevant information. A characteristic of dynamic morphology is that at each position, the outcome of applying the condition is not known before the operation. It is computed during the operation process using the image information within the region designated by the structuring element and the operation result on the neighborhood. This means that a dynamic morphological operation is an adaptive process and cannot be replaced by a straightforward transformation. Dynamic morphological operations are application dependent. Different conditions should be designed for different purposes. In this work, two dynamic dilations will be introduced and applied to closed boundary detection and moving objects detection which are two important and difficult problems in image processing.

The closed external boundary is an essential feature of an object. It conveys the shape of the object and can be used to extract the object from an image. Boundaries of objects are considered the most important part of image processing that links the original images and the interpretation [3]. Boundary detection often serves as an early stage in image understanding and other high level computer vision tasks. It is one of the most often used preprocessing steps in applications of recognition. In this work, the closed outmost boundary of an object is defined as a set of connected pixels which lie on the contour of the object. Since closed boundary represents the shape of an object and mathematical morphology deals directly with shapes in an image, morphology a reasonable approach to boundary detection. Most boundary detection methods are composed of two stages, detecting edges and linking the edge segments. It is a difficult task to connect those broken edges into meaningful boundaries and especially difficult to form the closed outmost boundary which outlines the silhouette of the object. The reason for this is that some edge segments do not correspond to meaningful boundaries and at the same time some boundary edges are missing. The proposed dynamic morphology leads to a new way of solving this problem.

Dynamic mathematical morphology is also used to detect moving objects in image sequences obtained from stationary cameras. Motion analysis can provide rich information of the surface structures of moving objects. Detecting and analyzing shape information of the moving objects in an image sequence may lead to a successful motion analysis. In this work the emphasis is on the study and analysis of applying dynamic mathematical morphology to detect shape information of moving objects. There are two kinds of image sequences according to the states of the cameras: the sequences obtained from motionless cameras and those from moving cameras. Only the first kind of image sequences is considered in this thesis. This technique has several potential applications such as video phone and video conference, surveillance, traffic control, customer counting. Basic idea of the newly proposed methods is to compare a pair of frames from an image sequence and get the moving points in each image. Dynamic morphology is then employed to get the shapes of the moving objects. Further analysis can be conducted on the basis of the geometrical information of each moving object in the image sequence.

The rest of the thesis is organized as follows, Chapter 2 introduces traditional mathematical morphology. Chapter 3 proposes dynamic morphology and then defines two dynamic morphological operations, expand dilation and roll dilation. In Chapter 4 a new boundary detection method is proposed on the basis of roll dilation. Chapter 5 introduces two moving object detection algorithms based on dynamic dilations. Chapter 6 gives the conclusions and suggestions for future work.

Chapter 2

Traditional Mathematical Morphology

The object of mathematical morphology is to provide a quantitative description of the geometrical structures in an image. A general morphological process is like probing and transforming an image with a structuring element which is a small pattern with a predefined shape. According to the image being processed, mathematical morphology can be divided into binary morphology and gray scale morphology. Both of them are now used widely. Mathematical morphology can be easily implemented in a parallel processing fashion. This chapter is a brief discussion about the traditional morphology.

2.1 Overview

Morphological operations are nonlinear translation-invariant transformations for images or other similar signals. They provide an efficient way for extracting shape information from an image and is used extensively. There are two basic steps in an image analysis process based on mathematical morphology: a geometrical transformation and then a measurement on the transformed image. Let **F** denote the image under study. According to the above procedure, a morphological image analysis operation consists of first a transformation T by a preselected structuring element **S**, followed by a measurement u. Examples of T are opening, closing, boundary extraction and skeletonization. The measurement $u[T(\mathbf{F})]$ is a number that can be a quality representation of weight, area, volume, etc. Thus, to obtain the quantitative shape information of an object such as size, spatial distribution, connectivity, convexity and orientation, this object is morphologically transformed and represented by different structuring elements. An appropriate measurement is then made on the transformation result. Object detection plays an important role in image processing. As the identification of an object, object features correlate directly with the object shape. Many high-level vision tasks depend on detecting these features in images. In terms of shape, a morphological operation can be described as a process which tends to simplify an image by preserving its essential shape properties and removing the trivial ones. For the work associated with morphological shape representations, please refer to [35] [36] [38] and [48] [2].

Mathematical morphology together with other logic and arithmetic operations can be applied to perform a great variety of image processing tasks. Moran [29] proposed a morphological transformation for edge enhancement. Sinha et al. [42] designed an algorithm based on morphology to recognize objects in a binary image. A skeletonization algorithm using gray scale morphology is presented by Shih and Mitchell [41]. More information about the recent development of mathematical morphology can be found in [27], [40] and [15]. Since in this work only binary morphology is used, the following discussions will be mainly on binary operations. There are two reasons that binary morphology is preferred to the grey level one. First the computation of binary morphological operations is simple. Secondly binary mathematical morphology has the ability to describe and process 2-D shapes of objects. In the rest of this Chapter, basic traditional morphological operations are introduced. These operations include *dilation, erosion, opening* and *closing* which are based on set theory. For the reason of consistence, the notations of mathematical morphology given by Haralick et al. [15] are referred. The definitions given by Serra [40] and Matheron [27] are slightly different.

2.2 Binary Morphology

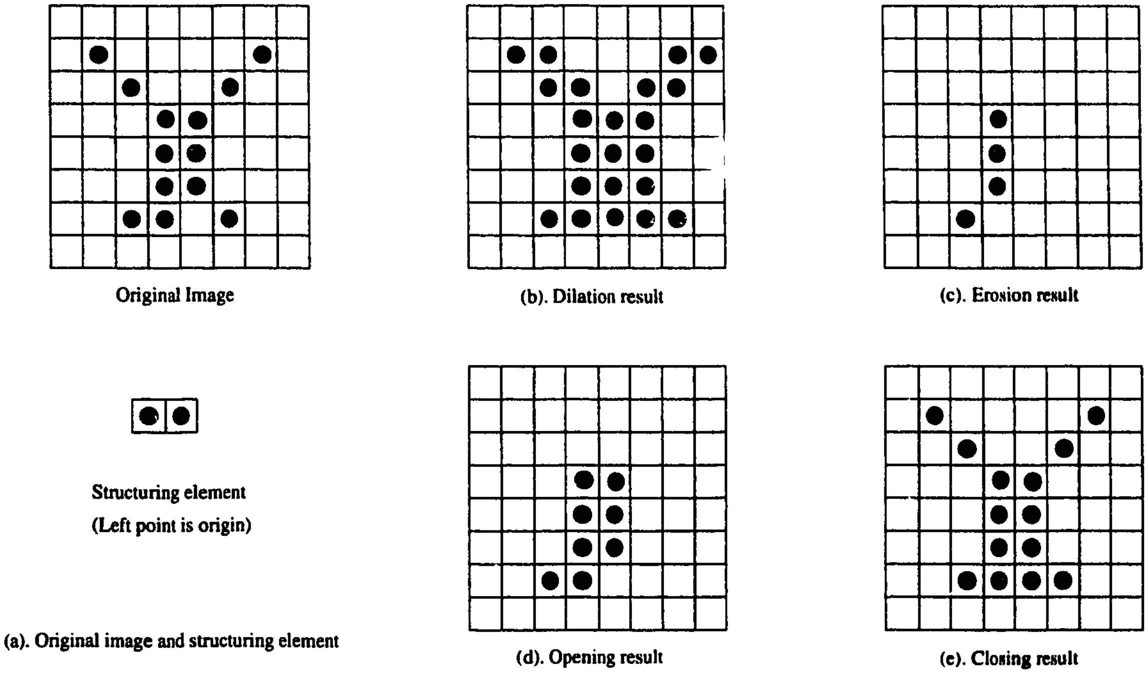
In image processing binary morphology is the mathematical morphology defined to operate on binary images. Different binary morphological operations have different affects on an image. Dilation expands object regions while erosion shrinks these regions. Closing fills small gaps within a pattern and opening deletes noise and insignificant details. A binary image can be considered as a set which is composed of the positions of all object pixels, the definition of mathematical morphology on binary images can be defined directly on this set. The following definitions are based on set theory. They may also be applied to signals other than binary images.

2.2.1 Dilation

Dilation combines two sets using vector addition of the set elements. If E^N denotes the N dimension Euclidean space, $\mathbf{F}$ and $\mathbf{S}$ are sets in N-space and $\mathbf{f}$ and $\mathbf{s}$ are their elements, $\mathbf{f} = (f_1, \ldots, f_N)$ and $\mathbf{s} = (s_1, \ldots, s_N)$, then traditional dilation of $\mathbf{F}$ by $\mathbf{S}$ is defined as

$$\mathbf{F} \oplus \mathbf{S} = \{\mathbf{x} \mid \mathbf{x} \in E^N,\ \mathbf{x} = \mathbf{f} + \mathbf{s} \text{ for } \mathbf{f} \in \mathbf{F} \text{ and } \mathbf{s} \in \mathbf{S}\}.$$

Applied to image processing, $\mathbf{F}$ is the object regions of a binary image and $\mathbf{S}$ is the structuring element. In the rest of the thesis $\mathbf{F}$ is usually called the image for the reason of simplicity except the cases that are stated explicitly. When we say "morphological operations on a binary image", we actually mean that the operations are performed on the object regions in the image. During a dilation process, the structuring element origin is put on each object pixel in the image and a pattern which is the same as the structuring element will be placed there. In real implementations, the result of $\mathbf{F} \oplus \mathbf{S}$ is usually computed by translating $\mathbf{F}$ with all points in $\mathbf{S}$ and then taking union of the translation results.



Original Image

(b). Dilation result

(c). Erosion result

Structuring element
(Left point is origin)

(a). Original image and structuring element

(d). Opening result

(e). Closing result

Figure 2.1: An example showing the concept of traditional binary morphology

Figure 2.1(b) is an example of dilation. The original image and the structuring element are shown in Figure 2.1(a). From the example it can be seen that all shapes in the original image have been extended by one pixel to the right. The gap between the two pixels at the bottom of the original image is filled up as it is smaller than the structuring element. The filled gap can not be recovered by other morphological operations. This means dilation is a non-reversible process. It is obvious that the dilation result is dependent on shape, size and the origin of the structuring element.

2.2.2 Erosion

Using the same notations as dilation, traditional erosion of **F** by **S** is defined as

$$\mathbf{F} \ominus \mathbf{S} = \{\mathbf{x} \mid \mathbf{x} \in E^N,\ \mathbf{x} + \mathbf{s} \in \mathbf{F} \text{ for every } \mathbf{s} \in \mathbf{S}\}.$$

Erosion of an image can be viewed as a process of finding all patterns that are the same as the structuring element in the image. The structuring element origin is placed at each pixel of the image during the operation. If all members of the structuring element fall inside the image the pixel is kept, otherwise it will be deleted. In another words, the result of $\mathbf{F} \ominus \mathbf{S}$ consists of all points $\mathbf{x}$ for which the translation of **S** by $\mathbf{x}$ fits inside **F**. In a real implementation the result is usually computed by translating **F** with the inverse of each point in **S** and then taking the intersection of the results.

Figure 2.1(c) is the erosion result of the image in (a). It shows that all regions in the original image have been shrunk by one pixel to the left. Details in the original image which are smaller than the structuring element have been removed. In this example these details include the two antenna like branches and the single pixel at the right bottom. This process is also non-reversible. In the followed opening operation, it is shown that the eroded details which are smaller than the structuring element can not be recovered in the dilation process using the same structuring element.

2.2.3 Opening and Closing

Opening and closing are defined directly on dilation and erosion. Using the definitions of dilation and erosion, opening and closing are defined as

$$\mathbf{F} \circ \mathbf{S} = (\mathbf{F} \ominus \mathbf{S}) \oplus \mathbf{S}$$

and

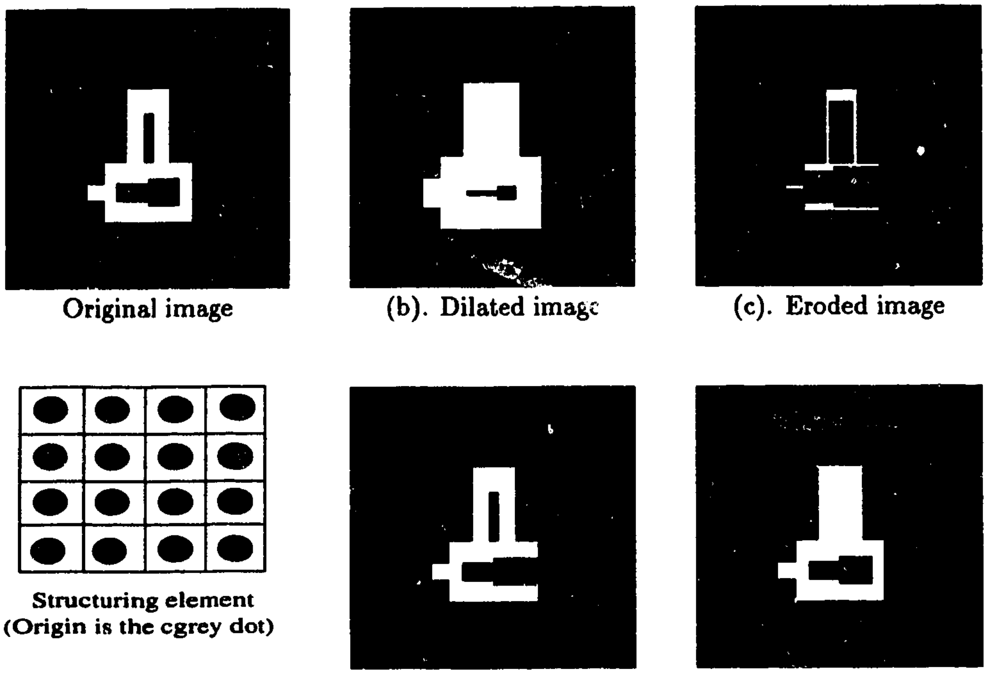
$$\mathbf{F} \bullet \mathbf{S} = (\mathbf{F} \oplus \mathbf{S}) \ominus \mathbf{S}.$$

In practice, dilation and erosion are sometimes employed in pairs because together dilation and erosion will eliminate specific image features which are smaller than the structuring element and keep those important global properties. In binary image processing, closing fills small gaps or holes within a pattern. Opening deletes noise and details of the image which are smaller than the structuring element.

Image transformations using iterative dilation and erosion or erosion and dilation are idempotent. This is an interesting property which means that the reapplication of opening or closing does not change the previous opened or closed result. This property has the practical importance in that it comprises complete and closed stages of image analysis algorithms. The reason is that shapes are naturally described in terms of the structuring elements by which these shapes can be opened or closed and at the same time remain unchanged. Morphologically filtering an image by an opening or a closing operation corresponds to the non-realizable ideal bandpass filters of conventional linear filtering. Once an image is ideal bandpass filtered, further ideal bandpass filtering will not affect the result.

Figure 2.1(d) shows the opening result of the image in (a). The details which are smaller than the structuring element have been removed in the opening operation. Only the shape features that are not smaller than the structuring element are left. This can be viewed as a smoothing operation of the original image. By selecting an appropriate structuring element, noise or small details in the image can be removed. Figure 2.1(e) is the example of closing which shows that the gaps smaller than the structuring element in the original image have been filled up. Closing also has the ability to smooth an image.

Figure 2.2 is an example of applying basic binary morphological operations to a block image. This example illustrates the feature of each operation more clearly.



Original image

(b). Dilated image

(c). Eroded image

(a). Original image and structuring element

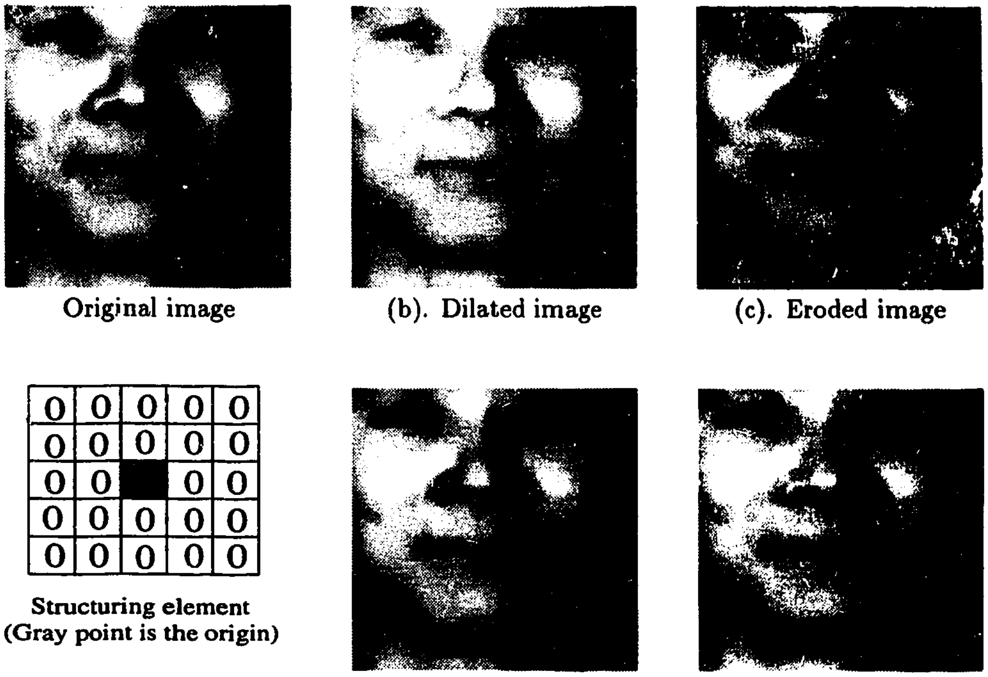
(d). Opened image

(e). Closed image

Figure 2.2: An example of basic binary morphological operations

2.3 Gray Scale Morphology

The above discussions are about binary images. In the case of gray scale images, the same concepts apply but the definitions are different from those of the binary morphology. Gray scale mathematical morphology is based on the definitions of top surface and umbra which are also related with set theory. The details will not be given in this thesis but can be found in [15]. Figure 2.3 is an example of gray scale morphology. It displays the results of performing the four basic morphological operations on a human face image. The structuring element is a small 5×5 plane which is shown in Figure 2.3 (a) along with the original image. It is clear that bright regions have been expanded and the result image look brighter after dilation. This is because maximum of intermediate images is used in gray scale dilation. Erosion makes an inverse effect on the image since it uses a minimum operation instead. Black areas in the original image have been enlarged after erosion. This makes the whole image looks darker. Opening and closing smooth the image. Sharp changes in the original image have been moderated in both the opened and the closed images.



Original image

(b). Dilated image

(c). Eroded image

0	0	0	0	0
0	0	0	0	0
0	0	■	0	0
0	0	0	0	0
0	0	0	0	0

Structuring element
(Gray point is the origin)

(a). Original image and structuring element

(d). Opened image

(e). Closed image

Figure 2.3: An example of basic gray scale morphological operations

Chapter 3

Dynamic Morphology

From the definitions in Chapter 2, it can be concluded that in traditional mathematical morphology each operation is applied to the whole of an image. During the operation all pixels of the image are processed one after another in a regular order which is usually the scanning sequence. But in many cases features or objects of interest merely occupy a small area of an image. There are also some application in which it is necessary to only process specific features or regions and thus prevent the result from being affected by other properties of the image. *Dynamic mathematical morphology* is proposed for this type of applications. In this chapter the concept of dynamic morphology is introduced. It is followed by the definition of binary dynamic dilation which is proposed on the basis of the traditional one. Two specifically designed dynamic dilations, expand dilation and roll dilation, are discussed in detail.

3.1 Concept of Dynamic Morphology

Dynamic morphology is a different concept from the traditional one. The prefix "dynamic" means that the operation is performed not in a regular order as in the traditional morphology, but following a specific sequence. Dynamic morphology only operates on the interested pixels and reacts to certain properties. It depends on a condition to trace objects or features of the image and to decide the path of the operation dynamically. During the operation, this condition is tested at each possible position of the structuring element origin, using the previous result and information of the image region covered by the structuring element. The output of applying the condition at each position can not be decided before the operation. The condition can only be tested during the operation process. In dynamic morphology, the condition can be specifically designed to make the structuring element only go though those pixels, regions or features that are required to be processed and thus reduces computational cost and increases precision of the result.

Dynamic morphology can be used in closed boundary or contour detection, object recognition, moving object detection and other imaging applications. The following discussions are focused on binary dilations which are used in two applications and proved to be powerful. All definitions can be extended to gray level images with some modifications.

3.2 Binary Dynamic Dilation

To save computation time and keep the result from being affected and degraded by irrelevant information, ***binary dynamic dilation*** is defined to operate only on specific regions or features in an image.

Using the same notations as traditional dilation, ***dynamic dilation*** of $\mathbf{F}$ by $\mathbf{S}$ under a condition C is defined as

$$\mathbf{F} \oplus_{D}^{C} \mathbf{S} = \cup_i \{\mathbf{x} \mid \mathbf{x} \in E^N,\ \mathbf{x} = \mathbf{o} + \mathbf{s}\ \ \forall \mathbf{o} \in \mathbf{O}_i \text{ and } \mathbf{s} \in \mathbf{S}\}$$

And we have $\mathbf{O}_{i+1} = Y_C(\mathbf{O}_i, \mathbf{F}, \mathbf{S})$. Again this definition is defined on the basis of set operation. Applied to image processing, $\mathbf{F}$ is a binary image and $\mathbf{S}$ is a structuring element. In the above definition condition C is a control mechanism which is used to direct the dilate operation. It is composed of constraints on selecting the position of the origin of the structuring element. Set $\mathbf{O}_i$ contains all valid positions of the structuring element origin at the ith step. Y_C is a function based on condition C. It calculates the next position set $\mathbf{O}_{i+1}$ from $\mathbf{O}_i$ and the relevant information in $\mathbf{F}$ under the restriction of C. Set $\mathbf{O}_0$ contains the start position of the dilation. Let $C(\mathbf{p})$ be a boolean function based on condition C:

$$C(\mathbf{p}) = \begin{cases} TRUE & \text{if } \mathbf{p} \text{ satisfies } C, \\ FALSE & \text{otherwise.} \end{cases}$$

Then each element $\mathbf{p}$ of $\mathbf{O}_{i+1}$ should satisfy $C(\mathbf{p}) = TRUE$. During the dilation process, condition C is tested at each potential position of the structuring element origin. The test is carried out on the basis of data given by image pixels overlaid by the structuring element and the existing operation results of neighbor pixels. Output of the test is used to guide the structuring element to move only along the pixels that are of interested objects or features. The selected pixels are those that satisfy condition C. The condition is compounded into the dilation process adaptively. This characteristic means that dynamic morphology is not equal to a one step transformation. Dynamic dilation is usually applied to the output image of a preprocessing stage. This intermediate image should contain the information that will be needed by condition C. Depending on applications and images to be processed the condition and its complexity vary greatly.

Other binary dynamic morphological operations such as dynamic erosion, dynamic opening and dynamic closing can be defined in a similar way. With these definitions, morphological operations can be carried out only on the parts of interest in an image. This may improve the result while reducing the computational expense.

3.3 Two Dynamic Dilations

In this section two dynamic dilations, *roll dilation* and *expand dilation*, are proposed. Conditions of these two operations are mainly based on the edge map of an object in which the boundary edges are broken. Expand dilation may also use data of the original image as a part of its condition. Roll dilation and expand dilation will be used in closed boundary detection and moving object detection which are two important and difficult problems in image processing.

Suppose there is a single surface object in an image and after edge detection the object is outlined by scattered edges along its boundary. To simplify the problem, it is assumed that the edge points are located evenly along the boundary of the object and there are no other edges either inside and outside the object. Figure 3.1 (a) shows an example in which broken edge segments disseminate along the boundary of a plane square. Figure 3.1 (b), (c) and (d) will be discussed later. Dynamic dilation is then designed to detect the object. Two approaches will be elaborated, one is to dilate from outside the object to enclose it and the other is to do the dilation inside the object until it is filled fully by the structuring element. These two approaches correspond to two dynamic dilations.

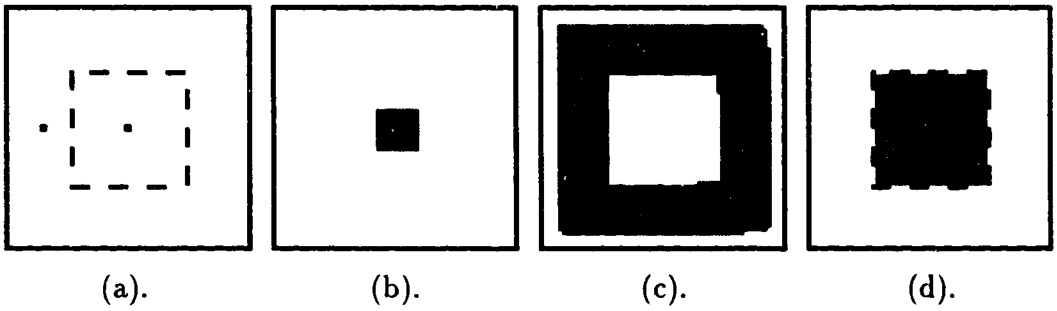

(a). (b). (c). (d).

Figure 3.1: An example of dynamic dilation

3.3.1 Roll Dilation

Roll dilation is defined on the basis of the above assumptions and a start point outside the object. It will be an ideal solution if an operation can only operate on boundary edges of the object, connect those broken edges and form a closed structuring element region surrounding the the object. In this way the object shape can be easily obtained from the dilation result. Roll dilation is designed for this purpose and the process of roll dilation can be described in the following way. During the operation a structuring element "rolls" along the outmost boundary edges of an object to form a coat outside it. It is already assumed that the boundary of the object has gaps between some neighboring edge segments. If the largest gap on the object boundary is smaller than the structuring element, these gaps can be filled by the structuring element and thus a smooth and connected boundary of the object can be produced.

Roll dilation should start from a seed point outside the object. The origin of the structuring element is first placed at the start point. A specifically designed condition is used to direct movement of the structuring element. To begin the discussion, condition C for roll dilation is simply described as *the structuring element stays outside the object and touches some edge pixels.* In the roll dilation process, each position of the structuring element origin should satisfy this condition. There are some problems of the above condition that need to be solved to make it work properly. The most important point is how to force the rolling operation to proceed along the object boundary. It is possible that either the structuring element always stays at a position, or it only rolls around a small edge segment of the boundary. Another concern is to keep irrelevant pixels from being considered. This can be satisfied by keeping the connectivity of the structuring element origin to its previous position.

The above condition C can make the structuring element touch boundary edges of the object, but cannot prevent it from staying at a point or rolling around a small edge fragment. To keep the structuring element moving along the object boundary, a more precise and powerful condition is needed. The new condition employs a cost function which is based on an age counter associated with each edge pixel and a history record for each non-edge pixel.

Suppose the original image $\mathbf{F}$ is defined on image plane $\mathbf{G} = \{(i,j) \mid i = 1,\ldots,M; j = 1,\ldots,N\}$ with height M and width N. After edge detection, edge pixels will form a set $\mathbf{E}$. Suppose $\mathbf{N_x}$ is the set of neighbors of point $\mathbf{x}$. Structuring element $\mathbf{S}$ consists of two parts:

$$\mathbf{S}_I = \{\mathbf{s} \mid \mathbf{s} \in \mathbf{S} \ \text{ and } \mathbf{N_s} \subset \mathbf{S}\}$$

and

$$\mathbf{S}_B = \mathbf{S} - \mathbf{S}_I.$$

For each edge pixel $\mathbf{x} \in \mathbf{E}$, an *age* counter $A(\mathbf{x})$ is defined as the number of times it is touched by $\mathbf{S}_B$ in the dynamic dilate operation. For each non-edge pixel $\mathbf{x}$, a *history* counter $H(\mathbf{x})$ is defined as the number of times that the origin of $\mathbf{S}$ resides at $\mathbf{x}$. After each dilation, $\mathbf{S}$ needs to be moved from the current position $\mathbf{o}$ to the next position which is selected from set $\mathbf{L_o}$:

$$\mathbf{L_o} = \{\mathbf{p} \mid \mathbf{p} \in \mathbf{N_o} \cap \mathbf{G} \text{ and } \ (\mathbf{S}_B + \mathbf{p}) \cap \mathbf{E} \neq \phi \text{ and } (\mathbf{S}_I + \mathbf{p}) \cap \mathbf{E} = \phi\}$$

where ϕ is an empty set. This means a new position for the $\mathbf{S}$ origin must satisfy the following three conditions:

1. It is a neighbor of $\mathbf{o}$.
2. When $\mathbf{S}$ resides on this point, $\mathbf{S}_B$ touches at least one edge pixel.
3. When $\mathbf{S}$ resides on this point, $\mathbf{S}_I$ does not cover any edge pixel.

To choose the next position from $\mathbf{L_o}$, a formula is designed to calculate the *cost* of moving to each candidate and the least costly one will be selected. The general rule for the cost function is that it should reflect the *wear* degree of a candidate position. The more a position is worn, the higher its cost will be. The degree of "wear" depends on the age of each edge pixel being touched and the frequency this position has been used. Thus the cost function is quantitatively defined by three factors, the total age of the touched edge pixels, the number of new edge pixels among the touched ones and the history of the position. At each candidate $\mathbf{p} \in \mathbf{L_o}$, the total age of the edge pixels that would be touched by $\mathbf{S}_B$ is calculated as

$$a_{\mathbf{p}} = \sum_{\mathbf{x} \in \mathbf{E} \cap (\mathbf{S}_B + \mathbf{p})} A(\mathbf{x}).$$

It is the basic cost of position $\mathbf{p}$. Of the touched edge pixels, each new one is given a bonus which reduces the cost by a certain amount. The cost is then adjusted to

$$c'_{\mathbf{p}} = a_{\mathbf{p}} - b\,k + b\,l$$

where b is a big constant associated with the size of $\mathbf{S}$, k is the number of new edge pixels touched by $\mathbf{S}_B$ and l is the size of $\mathbf{S}_B$, i.e.

$$k = ||\{\mathbf{x} \mid \mathbf{x} \in \mathbf{E} \cap (\mathbf{S}_B + \mathbf{p}) \text{ and } A(\mathbf{x}) = 0\}|| \quad \text{and} \quad l = ||\mathbf{S}_B||.$$

This guarantees to give a lower cost to a position which will let $\mathbf{S}_B$ touch more new edge pixels. The item $b\,l$ is used to shift the cost so that $c'_{\mathbf{p}} \geq 0$. If the position has never been used before, the corresponding cost is cut by scale b. Otherwise, a penalty $bH(\mathbf{p})$ will be added to the cost. The aim of doing this is to encourage $\mathbf{S}$ to move to the least used position. When all candidate positions have the same $c'_{\mathbf{p}}$, this adjustment will assign the lowest cost to the least used one. Finally the cost of selecting $\mathbf{p}$ is calculated as

$$c_{\mathbf{p}} = \begin{cases} c'_{\mathbf{p}}/b & \text{if } H(\mathbf{p}) = 0, \\ c'_{\mathbf{p}} + bH(\mathbf{p}) & \text{if } H(\mathbf{p}) > 0. \end{cases}$$

This cost function provides a mechanism to prevent the structuring element from rolling back to an old position. Among all the candidates, point $\mathbf{p}_s$ with the lowest cost will be selected as the next position of **o**, i.e. it has to satisfy the condition

$$\mathbf{p}_s \in \mathbf{L_o} \text{ and } c_{\mathbf{p}_s} \leq c_{\mathbf{q}} \quad \forall \mathbf{q} \in \mathbf{L_o}.$$

Then let $\mathbf{c} = \mathbf{p}_s$ and the age of each edge pixel touched by $\mathbf{S}_B$ and the history of the new position are updated by

$$A(\mathbf{x}) = A(\mathbf{x}) + 1 \quad \forall \mathbf{x} \in (\mathbf{S}_B + \mathbf{o}) \cap \mathbf{E}$$

and

$$H(\mathbf{o}) = H(\mathbf{o}) + 1.$$

This selection procedure favours new positions and **S** would touch as many new edge pixels as possible. The roll dilation process is over after it has enclosed the external boundary of an object. This is determined by wither the structuring element origin returns to the start position or it reaches image margins twice.

Figure 3.2 shows an example of cost calculation and new position selection in a roll dilation process. In this figure each edge pixel is represented by a square with the age inside. Each new candidate position of **o** is represented with two rectangles showing the history on the top and the cost on the bottom. The letter at the top-right corner of each image indicates the order in the computation sequence. Trace of the structuring element is shown by the grey area and trace of the origin **o** is indicated by the shaded squares in the last image. Figure 3.1 (c) is the roll dilation result of the square object used for the assumptions. The small black dot near the left image margin in (a) is the start point. The black square in (b) is the structuring element.

The modified condition C has three parts and is summerized as:

C_1: $\mathbf{p} \in \mathbf{N_o} \cap \mathbf{G}$.

C_2: $(\mathbf{S}_B + \mathbf{p}) \cap \mathbf{E} \neq \phi$ and $(\mathbf{S}_I + \mathbf{p}) \cap \mathbf{E} = \phi$.

C_3: $c_{\mathbf{p}} \leq c_{\mathbf{q}} \ \forall \mathbf{q} \in \mathbf{L_o}$.

For each new position $\mathbf{p}$ of the the structuring element origin $\mathbf{o}$, all the above sub-conditions should be satisfied. Function $\mathbf{O}_{i+1} = Y_C(\mathbf{O}_i, \mathbf{G}, \mathbf{S})$ can be detailed as

$$\mathbf{O}_{i+1} = \{\mathbf{p} \mid \mathbf{o} \in \mathbf{O}_i,\ \mathbf{p} \in \mathbf{L_o},\ c_\mathbf{p} \leq c_\mathbf{q}\ \forall \mathbf{q} \in \mathbf{L_o}\}$$

In roll dilation, $\mathbf{O}_i$ contains a single point. For other dynamic morphological operators, $\mathbf{O}_i$ may have multiple points. Set $\mathbf{O}_0$ contains the start point which can be determined by searching from the image margin.

A condition function $C(\mathbf{p})$ can be defined on the basis of the three sub-conditions for the purpose of easy use. At a position $\mathbf{p}$, if condition C_i is satisfied then $C_i(\mathbf{p}) = TRUE$. The value of $C(\mathbf{p})$ is logical AND of the results of applying these three sub-conditions to $\mathbf{p}$, i.e.

$$C(\mathbf{p}) = C_1(\mathbf{p}) \text{ AND } C_2(\mathbf{p}) \text{ AND } C_3(\mathbf{p}).$$

In a real implementation, only neighbors of the present position of the structuring element origin are considered in the next move. Thus $C_1(\mathbf{p})$ is always true and

$$C(\mathbf{p}) = C_2(\mathbf{p}) \text{ AND } C_3(\mathbf{p}).$$

At each step, the next position is point $\mathbf{p}$ that satisfies $C(\mathbf{p}) = TRUE$. A general roll dilation process can thus be listed as:

1. Find a seed point outside the object.

2. Put the structuring element at the present position.

3. Consider each neighbor $\mathbf{p}$ of the present position of the structuring element origin. Calculate $C(\mathbf{p})$, if true move to $\mathbf{p}$.

4. Stop if return to the seed point or find no new place to move to, otherwise goto 2.

From the above analysis it can be concluded that roll dilation is an adaptive process. The path of the structuring element origin is determined by condition C which can only be tested during the operation. Thus this process cannot be replaced by a one step transformation operation of the image.

An example of roll dilation is shown in Figure 3.3 where (a) contains a synthetic edge image of a fish and a round structuring element, (b) shows the roll dilation result and (c) is the inside contour of the roll dilation trace in (b). As can be seen from the figure, the roll dilation process has filled the gaps on the boundary of the object and produced a smooth and closed curve. Roll dilation can form the closed external boundary of an object using broken boundary edges of the object. In the above discussion it is assumed that there is no noise edges outside the object. This is not true in real applications and usually pre-processing is needed to make the edge image suitable for applying roll dilation. A closed boundary detection method based on roll dilation will be discussed later. In that method, efforts are made to deal with the existence of noise edges.

3.3.2 Expand Dilation

Given the previous assumptions on the edge map and a seed point within the object, expand dilation is defined to fill inside the object and find the object shape. Expand dilation is a different approach from roll dilation in detecting the object. In the following discussions, the notations used for roll dilation will be followed. There are also three parts of the condition for an expand dilation process.

First sub-condition is the connectivity of the structuring element origin to its previous position, i.e.

$$\mathbf{p} \in \mathbf{N_o} \cap \mathbf{G}$$

where **o** is the present position of **S** origin and **p** is the next position. This will guarantee that the new result is always connected to the existing result. The second sub-condition is based on the edge map. During the expand dilation process the structuring element should not touch any edge, or

$$(\mathbf{S}+\mathbf{p})\cap\mathbf{E}=\phi.$$

The last one is a constraint using the object surface in the original image. It is required that the average gray value deviation within a neighborhood designated by the structuring element on the original image should be less than a predefined threshold T_d. The average gray value (mean) of all pixels within the image area covered by $\mathbf{S}$ is

$$m(\mathbf{o})=\frac{\sum_{\mathbf{x}\ \forall\mathbf{x}\in(\mathbf{S}+\mathbf{o})\cap\mathbf{G}} f(\mathbf{x})}{b}$$

where $b=||\mathbf{S}||$ is the number of points within $\mathbf{S}$ and $f(\mathbf{x})$ is the gray value of a pixel $\mathbf{x}$ within the region delimited by $\mathbf{S}$. Then the average gray value deviation of all pixels within the original image area which is covered by the structuring element can be calculated by

$$D_g(\mathbf{o})=\sqrt{\frac{\sum_{\mathbf{x}\ \forall\mathbf{x}\in(\mathbf{S}+\mathbf{o})\cap\mathbf{G}}(m(\mathbf{o})-f(\mathbf{x}))^2}{b}}$$

So the third sub-condition is $D_g<T_d$. Lastly the three part condition of expand dilation is:

C_1: $\mathbf{p}\in\mathbf{N_o}\cap\mathbf{G}$

C_2: $(\mathbf{S}+\mathbf{p})\cap\mathbf{E}=\phi$

C_3: $D_g<T_d$

Using only the first two parts of the condition is usually enough to produce a correct result. Sometimes the third part can be used to improve the result. Function $\mathbf{O}_{i+1}=Y_C(\mathbf{O}_i,\mathbf{G},\mathbf{S})$ for expand dilation can be described as

$$\mathbf{O}_{i+1}=\{\mathbf{p}\mid\mathbf{p}\in\mathbf{N_o}\cap\mathbf{G}\text{ and }(\mathbf{S}+\mathbf{p})\cap\mathbf{E}=\phi\text{ and }D_g(\mathbf{p})<T_d\ \ \forall\mathbf{o}\in\mathbf{O}_i\}$$

Similar to roll dilation, a logic function could be built from these three sub-conditions for easy use. Given a candidate position $\mathbf{p}$ of the structuring element origin $\mathbf{o}$,

$$C_i(\mathbf{p}) = \begin{cases} TRUE & \text{if } \mathbf{p} \text{ satisfies sub-condition } C_i, \\ FALSE & \text{otherwise.} \end{cases}$$

Then $C(\mathbf{p})$ is defined as

$$G(\mathbf{p}) = C_1(\mathbf{p}) \text{ AND } C_2(\mathbf{p}) \text{ AND } C_3(\mathbf{p}).$$

In a real implementation only neighbors of the structuring element origin are considered in each move. Thus $C_1(\mathbf{p})$ is always true. $C_3(\mathbf{p})$ can also be waived if the edge image is of fair quality. Then $C(\mathbf{p})$ can have the most simple form $C(\mathbf{p}) = C_2(\mathbf{p})$.

After expand dilation, a connected region is formed within the edge image. This region should represent the shape of the object. Figure 3.1 (d) shows the result of doing expand dilation inside the square object in (a). The black dot at the center in (a) is a seed point to start the operation. Figure 3.1 (b) is the structuring element.

The condition of expand dilation is also a dynamic one as it depends on the previous dilation result. Output of applying the condition to an pixel cannot be determined before the operation. Obviously it is impossible to perform the operation in a scanning order. There are two key factors to the success of an expand dilation process: the first is to decide a start point, and the second is to select a structuring element. Only when the process starts from inside the object and size of the structuring element is larger than any distance between two neighbor edge segments on the object boundary, it may produce correct result. This issue will be discussed in detail when expand dilation is used to detect multiple moving objects.

The conduct of expand dilation is based on the assumption that there is no edges inside the object. In real applications there are usually some internal edges which will affect the expand dilation result. If there are enough edges inside the object, the expand process may fail or the result may only fill part of the object. In such a case, expand dilation should be combined with some preprocessing and post processing methods to improve the result.

It is worth mentioning that structuring element shape makes a difference in the result of either roll dilation or expand dilation. The better a structuring element shape matches the object shape, the more precise the result will be. In Figure 3.1 the square structuring element matches the square object very well so that the two results are accurate. Figure 3.4 shows another example of doing roll dilation and expand dilation on the square in Figure 3.1 (a). This time the structuring element used is a disc as shown in (b). The results are shown in (c) and (d) respectively. Since the disc's shape does not match the square very well, there is some deformation in the results. However, when an object shape is unknown it is usually better to use an isotropic structuring element.

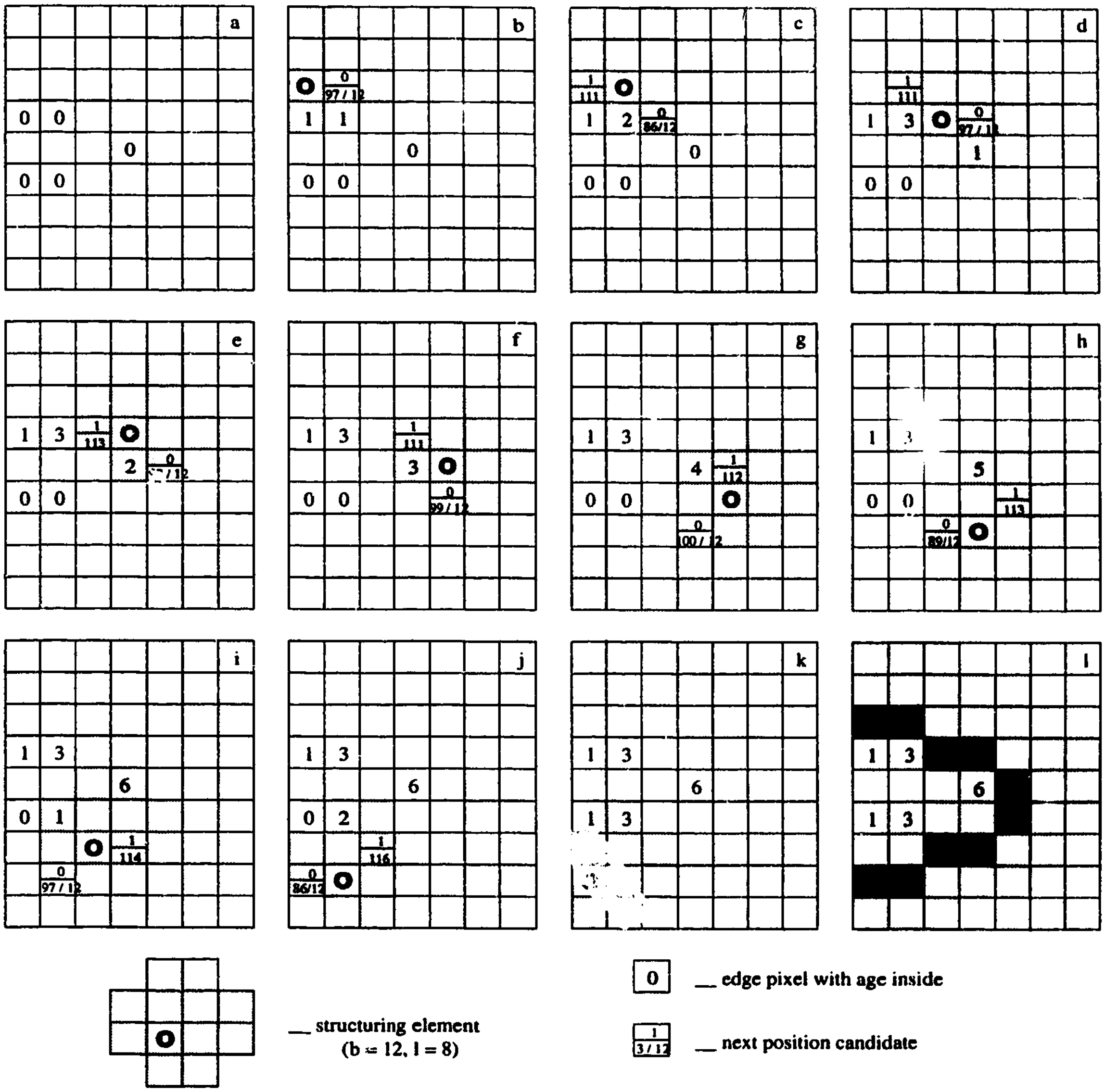
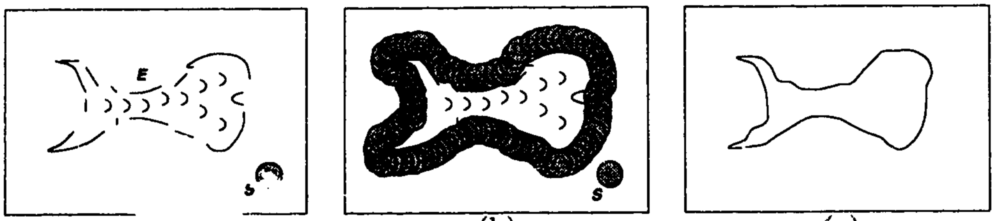
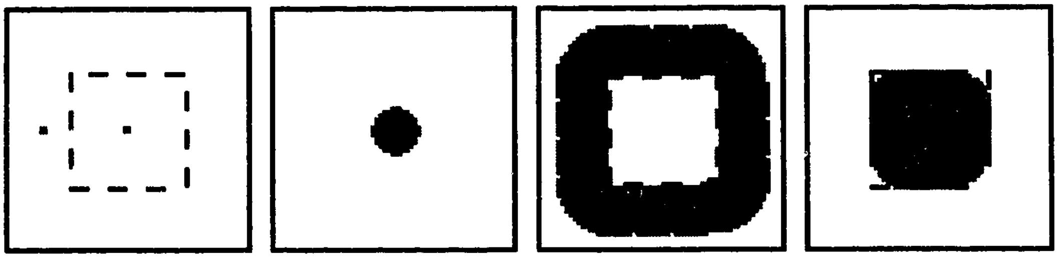

a
b
c
d
e
f
g
h
i
j
k
l
__ structuring element
(b = 12, l = 8)
0 __ edge pixel with age inside
1
3/12 __ next position candidate

Figure 3.2: An example of computing cost and selecting new position



((b) (c)

Figure [illegible] .nthetic fish image and its roll dilation result



(a). (b). (c). (d).

Figure 3.4: The effect of using a disc structuring element

Chapter 4

Boundary Detection Based on Roll Dilation

Since boundaries convey critical shape information of objects, boundary detection plays an important role in image processing. It often serves as an early stage in image understanding and other high-level vision applications. Boundary detection is also used extensively in medical image processing such as tumor diagnosis. In recent years many new algorithms have been proposed. Most boundary detection algorithms consist of two parts: edge detection and edge linking. It is a difficult task to connect broken edges into meaningful boundaries. Roll dilation provides a new approach to the solution of this problem and leads to an efficient boundary detection algorithm.

4.1 Related Work

Ballard and Brown [3] summarized some basic boundary detection methods based on Hough transform, graph searching, dynamic programming and contour following. From then on, more algorithms and different techniques have been developed to tackle the problem (eg. [37] [43] [8] [34] [39]). Among these techniques are neural network [17]. fuzzy clustering [9] and hierarchical model [26], just to name a few. While most methods handle static objects there are also algorithms proposed to deal with moving objects [11] [31]. Edge detection is the mostly used tool and edge quality is the key factor to the performance of many algorithms. Because some edges do not correspond to a object boundary and some boundary edges are missing. boundaries of an object are usually composed of broken edge segments. It is a difficult task to connect these broken edges into meaningful boundaries. In some applications the closed external boundary is of special interest as it outlines the 2-D shape of an object.

An algorithm based on contour following was designed by Lacroix [21]. This method has three modules. First a conventional edge detector is used to compute edge strength and edge orientation. The second step assigns a measurement of the possibility of being part of an edge to each pixel. In the last step a contour following method is used to follow those processed edge pixels. The process starts with the pixels which have a certain possibility of being an edge. A contour is followed as long as in the neighborhood of the end points of the contour, there is an edge pixel which satisfies certain requirements on the direction and the possibility of being an edge. After all possible candidates have been studied, short contours are removed by checking the length of each detected contour.

William and Shah [45] proposed an algorithm which employs edge following in a multi-scale space. Firstly the best partial contour is formed at the largest scale. Then at the next finer scale the neighborhood around the end points of the contour are examined to detect possible edge pixels with a direction similar to the one of the end points. The original contour is extended by the newly found edge. This procedure continues until a closed contour is formed or the smallest scale is used. At each scale, a possible edge pixel set is placed into a queue with the point having the largest magnitude on the top. The strongest point which is not a part of a contour is retrieved from the queue. Its neighbor point in the computed direction is examined first and then the point in the adjacent direction on each side of it. Each branch is followed to the end and at each point a weight is assigned to the the measurement of noisiness, the measurement of curvature, the contour length and the gradient magnitude. The point with the largest average weight is chosen. This course is then restarted from the initial point and proceed in the opposite direction.

In the boundary detection algorithm proposed by Jain [19], variable resolution (VR) technique is used to process edge segments and connect those broken edges. VR masks are designed for this purpose and each mask is composed of 9 square regions which are arranged in 3 rows and 3 columns. The center region is a 3×3 window. Each of the eight peripheral regions is also a 3×3 window but in a reduced resolution. Together these windows will cover a 9×9 neighborhood in an image. Firstly the image region covered by the center window is studied in detail. If certain conditions are satisfied indicating the existence of an edge, some of the 8 surrounding low resolution regions in the same direction are searched to extend the edge. In the case there is no edge in the center window, other windows will be looked at roughly. When an edge is detected, the center window is marked accordingly. The combination result is that the broken edges covered by the mask have been connected. This method produced very good experimental results.

These methods usually do not form the closed external boundary. Among other newly proposed algorithms, one kind is designed to detect external boundaries. It is based on active contour models which are also known as "snakes". An active contour is a deformable line modified by internal "forces" from the geometry of the curve and image forces due primarily to the intensity gradient. It was first introduced by Kass et al. [20] and some recent work can be found in [22] and [23]. Active models for boundary detection are surveyed by Menet et al. [28]. William and Shah [46] also proposed an active contour model which evolves by locally minimizing an energy function. The function is a weighted sum of three terms representing continuity, curvature and image forces. In each iteration, the function is evaluated for a current point and its $m \times m$ neighborhood. The point with the minimum energy will replace the current point. Active contour methods usually deal with simple shapes and require extensive computation to iteratively solve some energy equations. They also need an initial contour inside the object to start with. This means the object position and basic shape should be known before hand.

In the following discussions an algorithm is developed based on roll dilation to detect closed object boundaries from real scene images. In the rest of the chapter, "boundary" means closed external boundary if not explicitly stated.

4.2 Algorithm Description

A boundary detection algorithm is designed on the basis of edge detection and roll dilation. This method can be used to detect boundaries of more than one object in an image. Background of the image should be simple to allow a reasonable separation between any two objects. The detected edges of each object may be broken. The algorithm is robust within a certain noise level. It can be divided into three phases: *edge detection*, *morphological grouping* and *boundary formulation.* Figure 4.1 gives a general overview of these steps and a detailed description follows.

4.2.1 Edge Detection

Although any edge detection method can be used, the one introduced by Canny [7] is employed. In our experiments all parameters of the edge detection algorithm are fixed and changing these values only affects the results slightly. Noise edges can cause slight deformation of the detected boundaries. If some scattered edges fill the area between different objects with a certain density, these

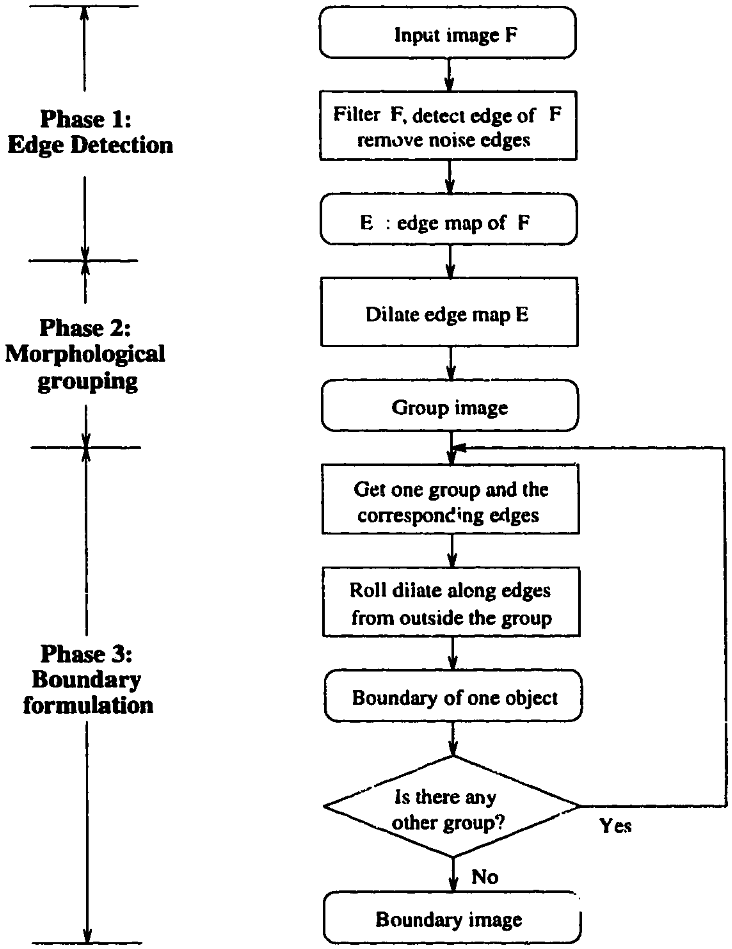


Figure 4.1: Structure of the boundary detection algorithm

objects may be detected as one object. Small edges near an object boundary can cause shape distortion of the object. Figure 4.2 shows an example in which the detected boundary of an artificial fish is deformed by some noise edges near the fish tail.

Removing these noise edges can improve the quality of the result. Two ways are employed in the algorithm. The first method is to apply a 3 × 3 median filter to the original grey level image. The second way is to remove noise edges from the edge map after edge detection. The most straightforward method is to delete

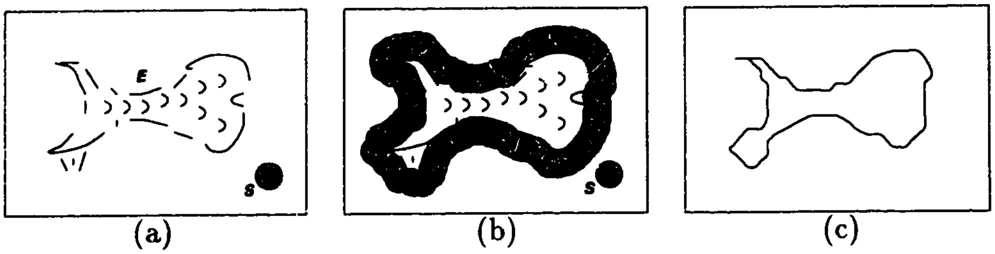


Figure 4.2: Noise edges deform the detected boundary of an object

isolated edge pixels and weak edges. If a suitable sized structuring element is used, requirement on the edge map quality can also be loosen.

4.2.2 Morphological Grouping

To provide a rough estimate of each object position and its area range, a simple morphological dilation is applied to the edge map. Most dilated edges will be connected into several regions. Each connected area is considered as a marker of a potential object. Using these connected regions, the detected edges can be separated into several groups with each one corresponding to a region. An example is shown in Figure 4.3 in which detected edges are connected into five groups by a dilation. Roll dilation from an outside point is then carried out on each edge group to get the outermost boundary of a potential object.

4.2.3 Boundary Formulation

Roll dilation is performed around each edge group to form the boundary of an object. The process is similar for all objects. It is already known that there are two key factors in roll dilation, the start point and the structuring element. To find a start point, a structuring element moves towards an edge group from an image margin. The first position where the structuring element touches some edges is used as the start point. Structuring element size plays an important role

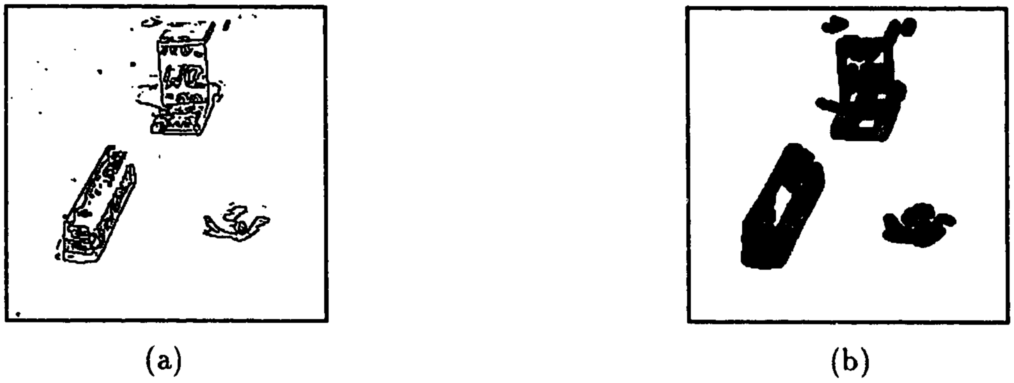


Figure 4.3: An example of morphological grouping

in roll dilation. It should be larger than any gap between two neighbor boundary edge segments. Generally speaking a large structuring element can be employed to form good boundaries in most cases. Since the shape of each object is not known, discs are used as structuring elements in all experiments. The structuring element then rolls along the outside edges of the object from the start point. As discussed before, there is an age counter associated with each edge pixel. The age counter will be incremented when an edge pixel is touched by the structuring element. Each time the structuring element moves forward, it will select one valid position from the neighbors of the present position. At a valid position only some of the structuring element margin points touch some edge pixels. For each valid position the cost is calculated and the one with the lowest cost is selected as the new position.

The termination condition of roll dilation depends on the position of the edge group in the image. If the edge group is totally contained in the image and the shortest distance from any edge pixel to any image margin is bigger than the radius of the structuring element, the structuring element will return to its start point after circling around the object and the roll dilation process is then over. Figure 4.4 (a) is such an example in which the lightly marked area indicates an object and the darker area is the roll dilation trace. Otherwise, the structuring element will bump one of the four image margins. In this case it will turn around and continue in the opposite direction until it comes to an image margin again. If the two touched margins are not opposite to each other, the object is defined to be either at one corner of the image and part of the object is out of the image from the corner, or on one side of the image and part of the object is out of the image from this side. Figure 4.4 (b) and (c) are examples of such cases. In these two cases the object is detected after the structuring element reaches image margins twice and the process is then terminated. If the two touched margins are opposite to each other as shown in Figure 4.4 (d) and (e), the object is defined as one that lies across the image and extends out of the image from the touched borders. In this case the roll dilation has only formed the boundary along one side of the object. To get the full boundary, roll dilation will be restarted from the other side of the image and stops until it reaches the margins twice. The above procedure continues until boundaries of all edge groups have been formed.

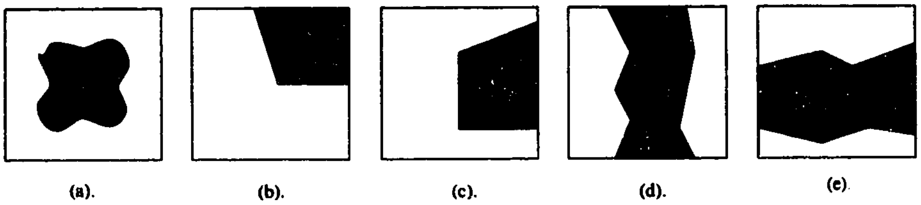


Figure 4.4: Different cases of boundary detection

In edge images there usually exit some splinter like lines connecting to object boundary edges. These "splinters" come from the background or the noise and will deform the roll dilation result. After the detection of each object, some of these edges can be removed if they have not formed regions. An edge pixel is said to be *isolated* if none of its neighbor pixels is a background pixel in the roll dilation result. All isolated edge pixels can be removed from the roll dilation result

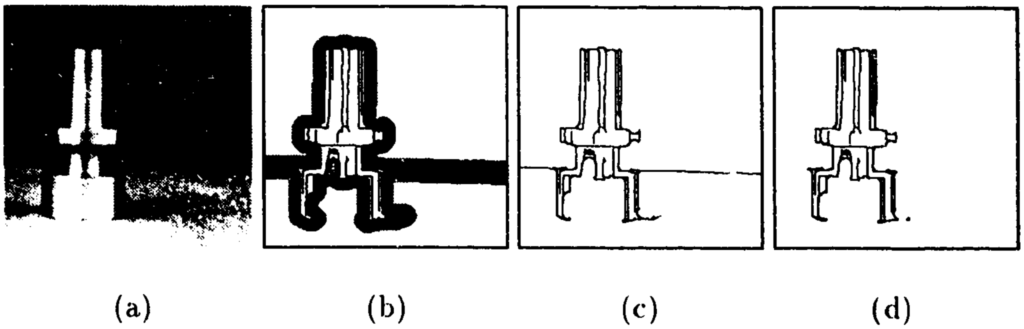


Figure 4.5: The effect of removing isolated edges in boundary detection.

to make the detected boundary smoother. An example is shown in Figure 4.5 in which (a) is the original image and (b) is the roll dilation result. The detected boundaries before and after removing isolated edges are displayed in (c) and (d). It is obvious that the boundary in (d) looks better than that in (c).

Pseudo Code

Finally the algorithm can be described using the following pseudo code:

```
BEGIN
    PHASE 1: edge detection
        INPUT image F
        FILTER F to remove random noise
        DETECT EDGE of F, E = edge map of F
        FILTER E to remove noise edges
    PHASE 2: morphological grouping
        DILATE edges in E
        GROUP edges in E by connected regions
    PHASE 3: morphological roll dilation
        FOR (;;)
            IF there is no more edge group in E
                EXIT
            ELSE
                SELECT one group $A$
                MARK all edges in $A$
                FIND a start point $(i_s, j_s)$ from outside $A$
                $i = i_s$, $j = j_s$
                FOR(;;)
                    FOR each neighbor of $(i, j)$
                        IF structuring element margin touches some edges
                            CALCULATE the cost
                        ENDIF
                    ENDFOR
            SELECT the neighbor $(i_n, j_n)$ with the lowest cost
            $i = i_n$, $j = j_n$
            IF $(i, j)$ reaches $(i_s, j_s)$ or reaches image boundary twice
                EXIT
            ENDIF
        ENDFOR
        CALCULATE the area of the detected region
        IF size of the region $>$ a predefined threshold
            COUNT it as an object
        ENDIF
    ENDIF
ENDFOR
```

4.3 Experimental Results

This method has been tested on several images and the test results are encouraging. Figure 4.6 to Figure 4.10 show five examples in which pictures are arranged in the same way. The original image and the detected edges are shown in (a) and (b), roll dilation result and detected external boundaries are given in (c) and (d) respectively. Figure 4.6 is an example on a test image with two objects. There is a chair and a lamp in the image. Due to the light reflection some part of the lamp cannot be separated from the background and the detected boundary is deformed at such place. Figure 4.7 also contains two objects, a ring and a cup. The boundary around the cup is formed correctly although the handle is not detected successfully in the edge image. Shadow of the two objects deforms the boundaries slightly. Figure 4.8 shows a case with three objects and the boundaries have been detected successfully. There are four objects in Figure 4.9 and the algorithm has detected the boundaries even when the edges are broken obviously. There are three objects in Figure 4.10. Although there is noise in the background, boundaries of these objects have been detected with only minor distortion in two of them.

This method is compared with the three module method of Lacroix, the multiscale method of William ed al. and the VR method of Jain. Figure 4.11 to

Fig:re 4.15 list the results of applying these four algorithms to five images. The original images and the results of the three module method, the multi-scale space method and the VR method are courtesy of Manoj Kumar Jain. All figures are arranged in the same way. The original image is shown in (a). The roll dilation result and the detected boundary with inside edges are listed in (b) and (c) respectively. The results of Lacroix's method, William's method and Jain's method are shown in (d), (e) and (f). It can be concluded from these figures that the results from the VR method are much better than Lacroix's method and William's method but sometimes the external boundaries are not closed. The proposed method always produces closed boundaries and meaningless edges around an object are usually removed. Since the code of the other two methods is not available, only VR method is applied to the images in Figure 4.7 to Figure 4.10. The results are shown in Figure 4.16.

4.4 Result Analysis

From the results in section 4.3 it can be concluded that the advantage of the proposed boundary detection algorithm is that it can form closed external boundaries of the objects in an image. The closed external boundary of an object is used to calculate different measurable properties of the object. These properties include area, perimeter, mass center. More complicated features such as convexity, curvature, corner points can also be calculated. These measures of an object may be applied to recognize the object.

By comparing the results of the roll dilation based method with that of Lacroix's method, William's method and Jain's method, it can be seen that there are many noise edges in the results of Lacroix's method. William's method removes most noises but at the same time some meaningful boundaries are also deleted. Jain's VR method produces much better results than the above two methods. Subjectively the results of the proposed method is similar to that of the VR method except that the detected external boundaries of objects are closed. As just mentioned, the closed external boundary can be used to calculate the measurable shape features of an object.

The quality of the detected boundary is dependent on the structuring element and the edge image. To quantitatively describe this effect several variables are introduced. Assuming A_r is the real size of an object, A_d is the area within the detected boundary and the object location is correct, *size correctness* is defined as

$$R_s = \max\{0, 1 - \frac{|A_d - A_r|}{A_r}\}.$$

This simple definition represents the quality of the boundary detection result with respect to the size of the object. R_s is close to 1 when a detected boundary fits the object boundary very well, and approaches 0 when the detected boundary is far away from the real one. In the experiments $|A_d - A_r|$ is usually less than A_r. Shape and size of the structuring element will affect the detection result. Only discs are used as structuring elements in the experiments presented here. In the following discussions *structuring element size* is defined as its maximal radius and represented using S_s.

Figure 4.17 shows the relationship between R_s and S_s for three images. Other images have produced similar curves. It can be seen from the figure that structuring elements with S_s larger than 2.5 pixels produce much higher R_s than those smaller ones. But R_s grows slowly after S_s has exceeded 2.5 pixels. The reason is that in roll dilation, when S_s is smaller than any gap on the boundary edges of an object, it will fall inside the object and the detected boundary will not fit the real object. A structuring element which is larger than all gaps on the object boundary can leap over these gaps. The corresponding dilation process rolls along the outmost edges of the object and the detected boundary will fit the real object. When S_s is increased further, the detected boundary will change only slightly. Another result of increasing S_s is that the structuring element will hit some noise edges in the background with greater likelyhood. The turning point of R_s curve, which occurs when $S_s = 2.5$ in Figure 4.17, can be used to determine an optimal structuring element size in terms of size correctness. For other test images the best structuring element size may be different.

Structuring element size also affects roll dilation time. Figure 4.18 shows the computation time C_t observed by testing three images using different sized structuring elements. It can be concluded that when S_s is increased, C_t drops first and then begins to increase after a certain structuring element size. The cause is that for small sized structuring elements, roll dilation may fall into the object due to broken edges on the object boundary. Sometimes the process will go around each edge segment and consume much computation time. When S_s reaches a certain level, the dilation process cannot fall through the edge gaps. The operation will go along the outside of the object and thus reduce computation time. Further increasing S_s will increase C_t. Computation time C_t is defined as $C_t = D_t \times L_n$ in which D_t is the dilation time at each position, and L_n is the total number of dilation operations. D_t can be expressed as $D_t = gS_s^2$ where g is a constant.

4.5 Summary

A new algorithm is proposed to detect object boundaries using roll dilation. This algorithm is robust when applied to certain images. The proposed boundary detection method has the following characteristics:

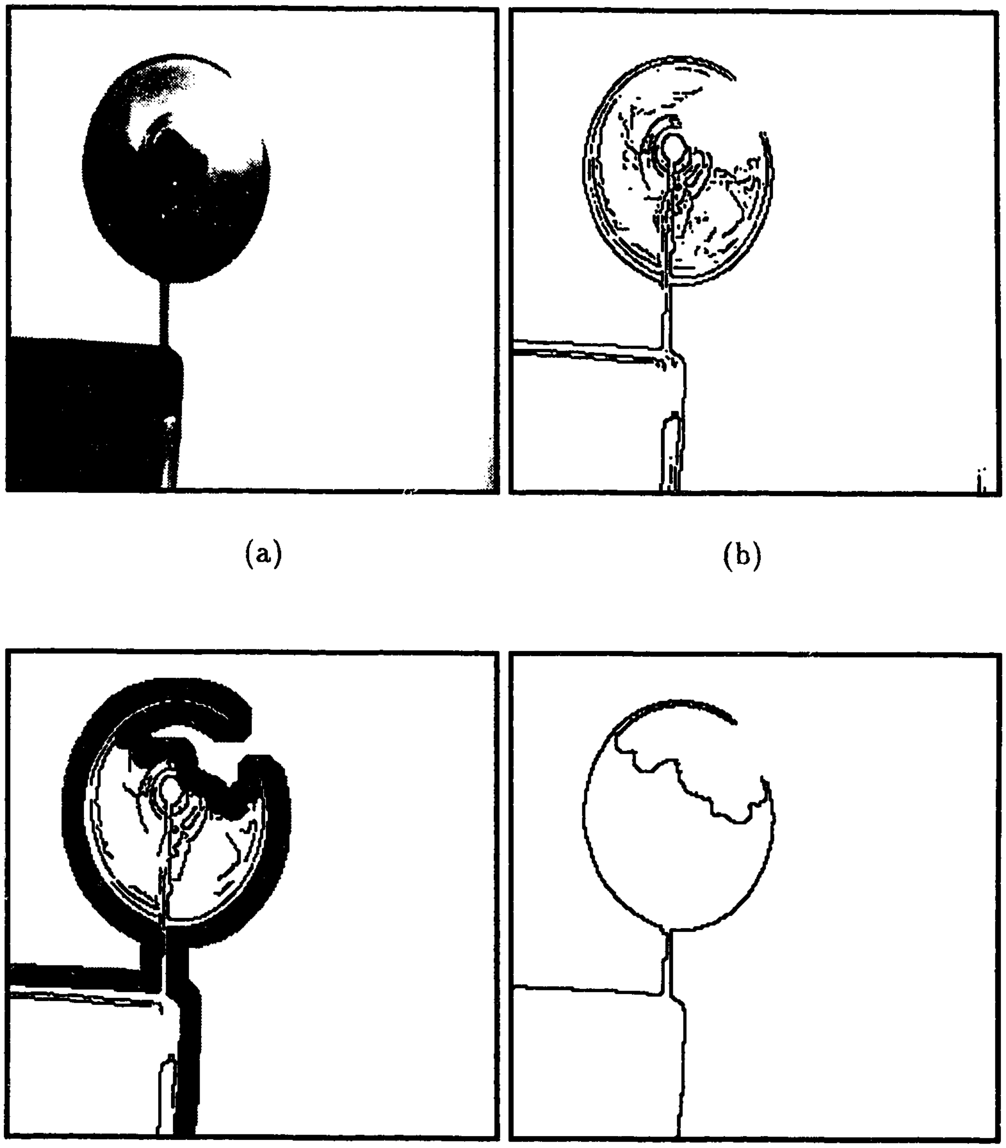
- Using roll dilation, implementation of the algorithm is simple.
- Fixed parameter values work for most image in both edge detection and roll dilation processes.

- It can process scenes with separate objects of various shapes and sizes in a simple background.
- Let the detected boundary length of an object be l, the total number of operations is

  $$t = l(n_n \times ||\mathbf{S}|| + ||\mathbf{S}|| = l(n_n + 1)||\mathbf{S}||,$$

  in which n_n is the number of neighbors to be searched at each position and $||\mathbf{S}||$ is the size of the structuring element. If the size of the image is $N \times N$, then in the worst case $l = 4N$ and $t = 4N(n_n + 1)||\mathbf{S}||$. Since usually we have $n_n << N$, the time complexity of the algorithm is $O(N||\mathbf{S}||)$.

This method can be improved in several ways. Firstly, the roll dilation process is now sensitive to noise edges. The object boundary will be deformed if there are noise edges along it. More complicated conditions might be combined into the roll dilation to make it more resistant to noise. This way, some noise removing steps could be simplified or even omitted. Secondly, additional processing techniques must be employed to handle a complex background. One way to clean the background is to subtract the complex background from the image. This can be used when the background image is already known or an image series is available to get an average background. As will be shown in Chapter 5, moving object detection environments can provide a method of cleaning the background. However in still images, certain knowledge is required to handle complex background.



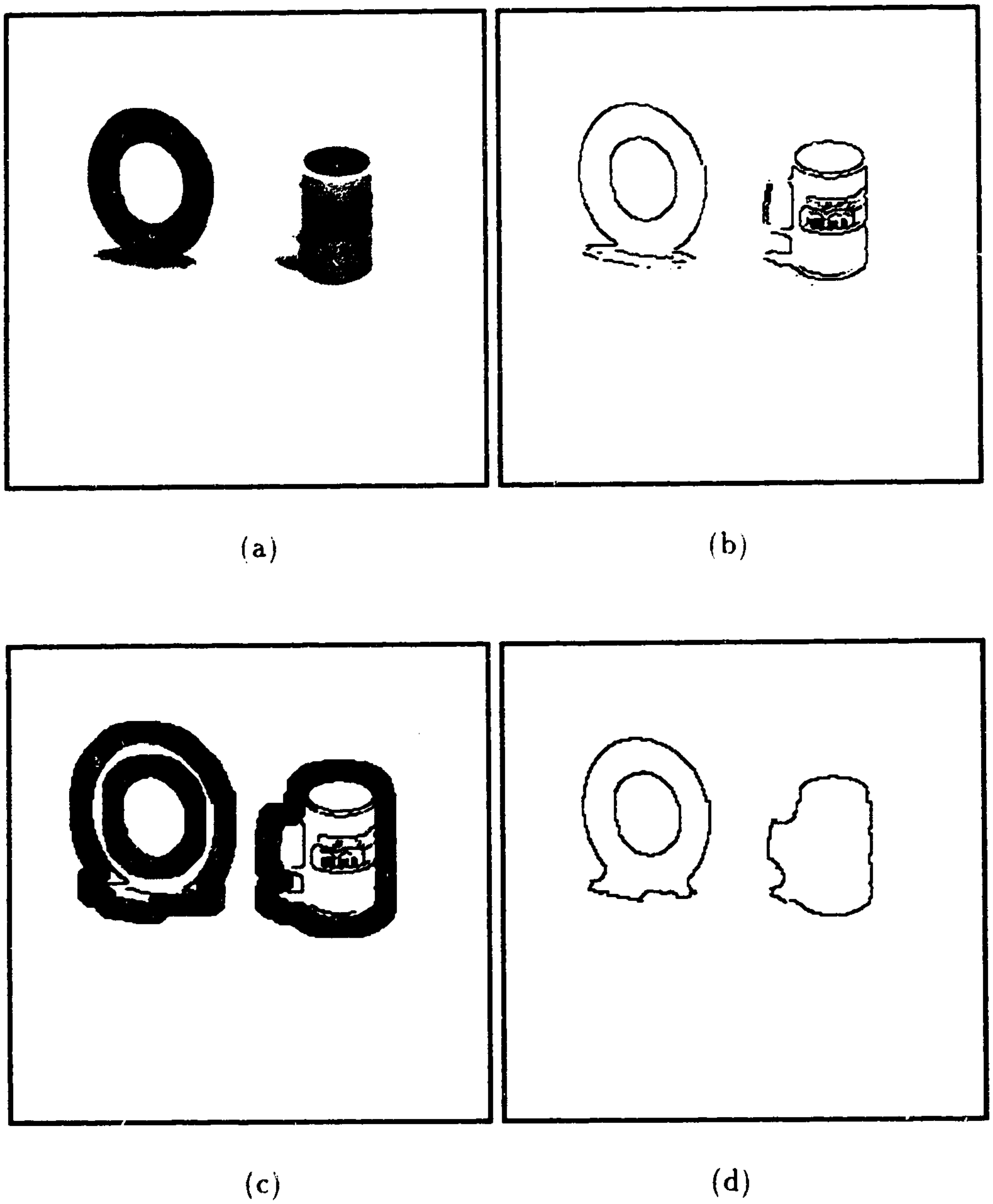
(a)

(b)

(c)

(d)

Figure 4.6: Boundary detection result on a test image with two objects



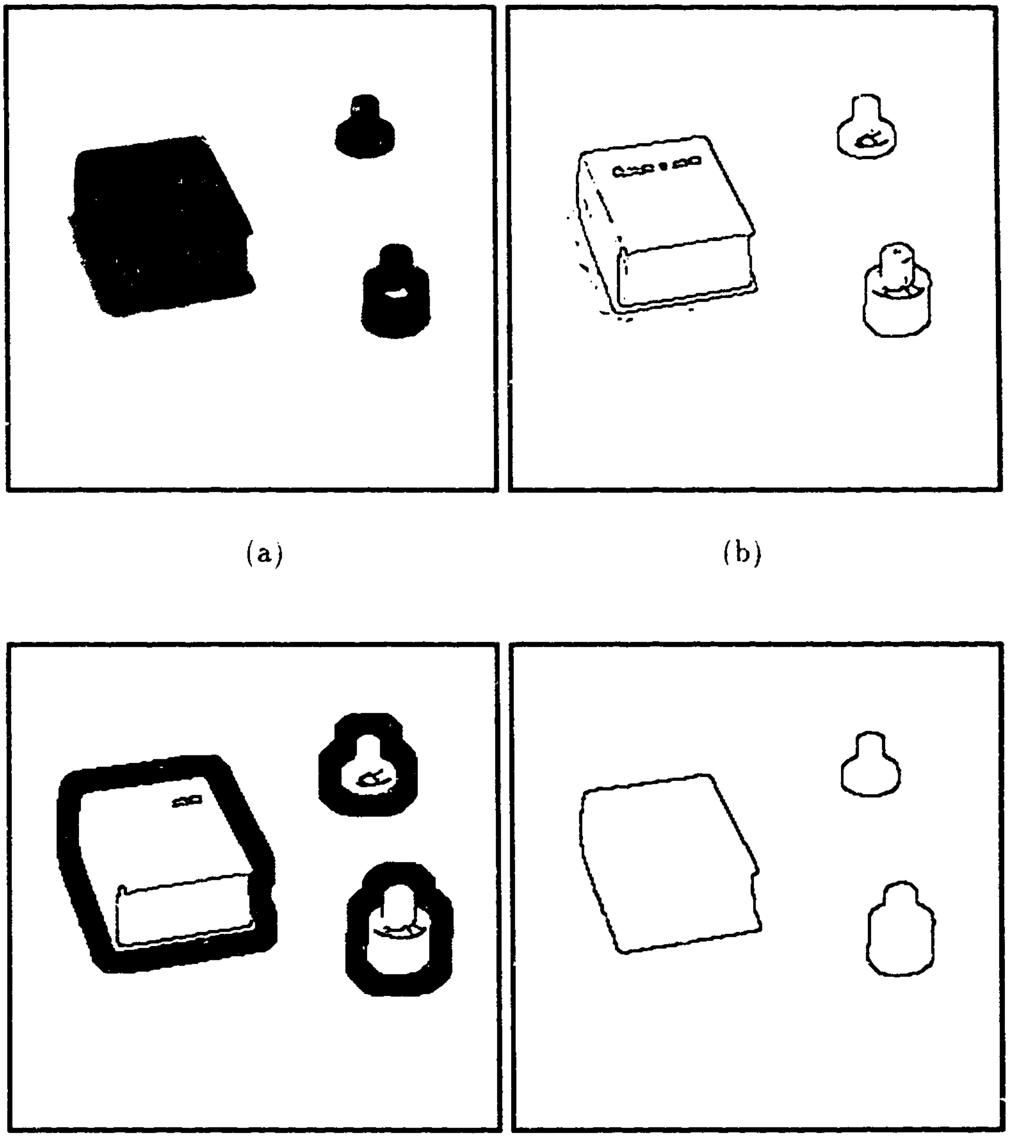
(a)

(b)

(c)

(d)

Figure 4.7: Boundary detection result on a test image with two objects



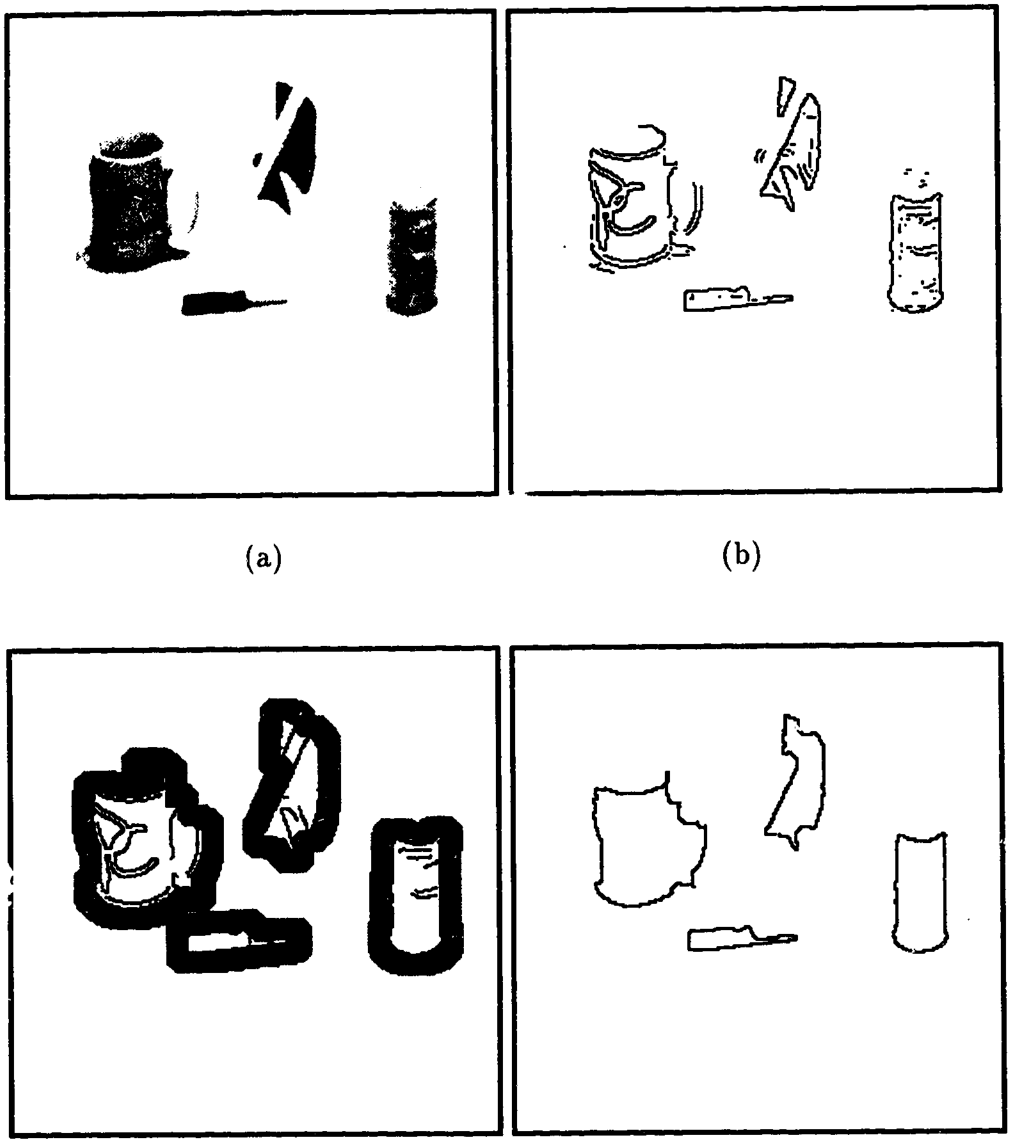
(a)

(b)

(c)

(d)

Figure 4.8: Boundary detection result on a test image with three objects



(a)

(b)

(c)

(d)

Figure 4.9: Boundary detection result on a test image with four objects

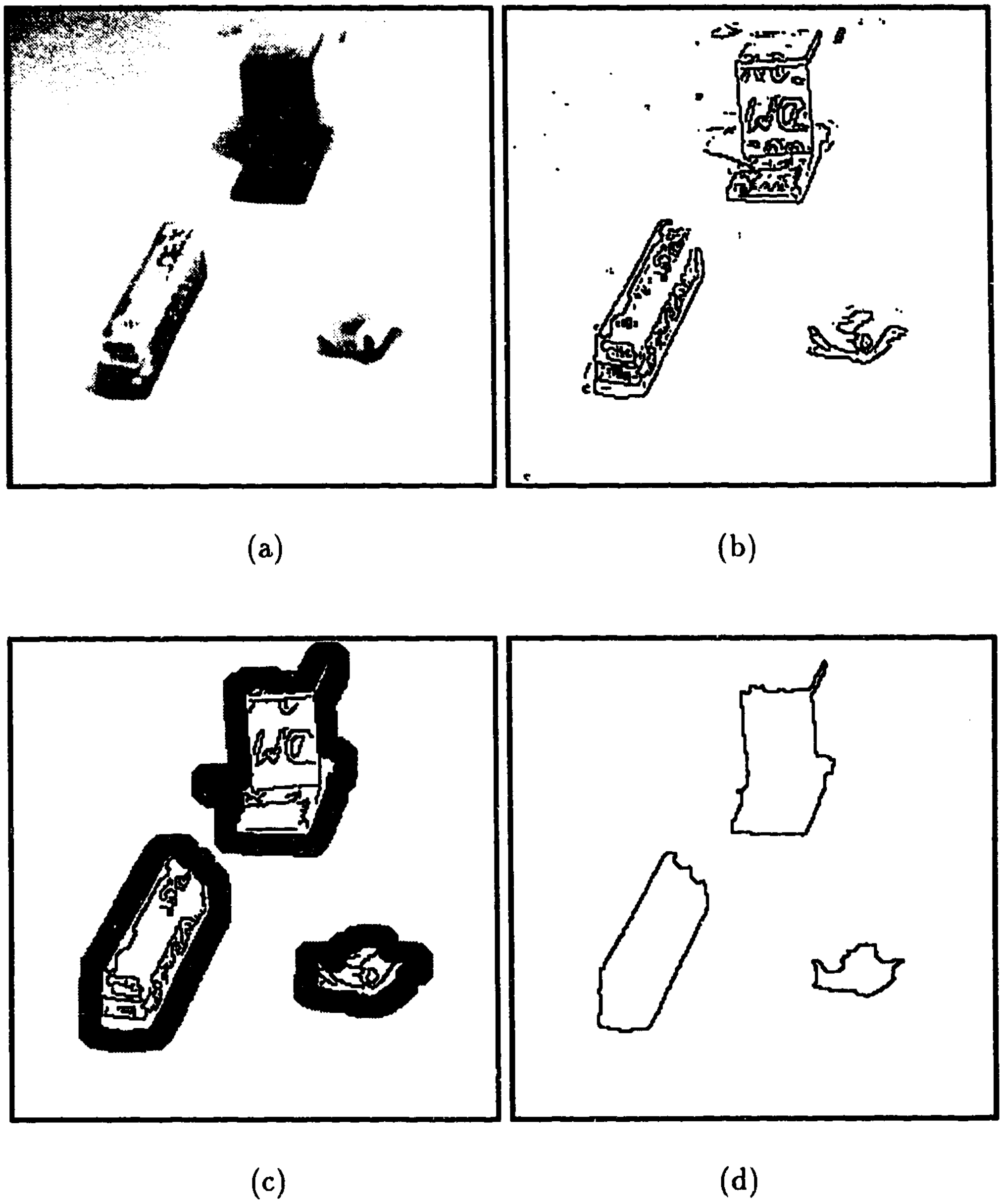
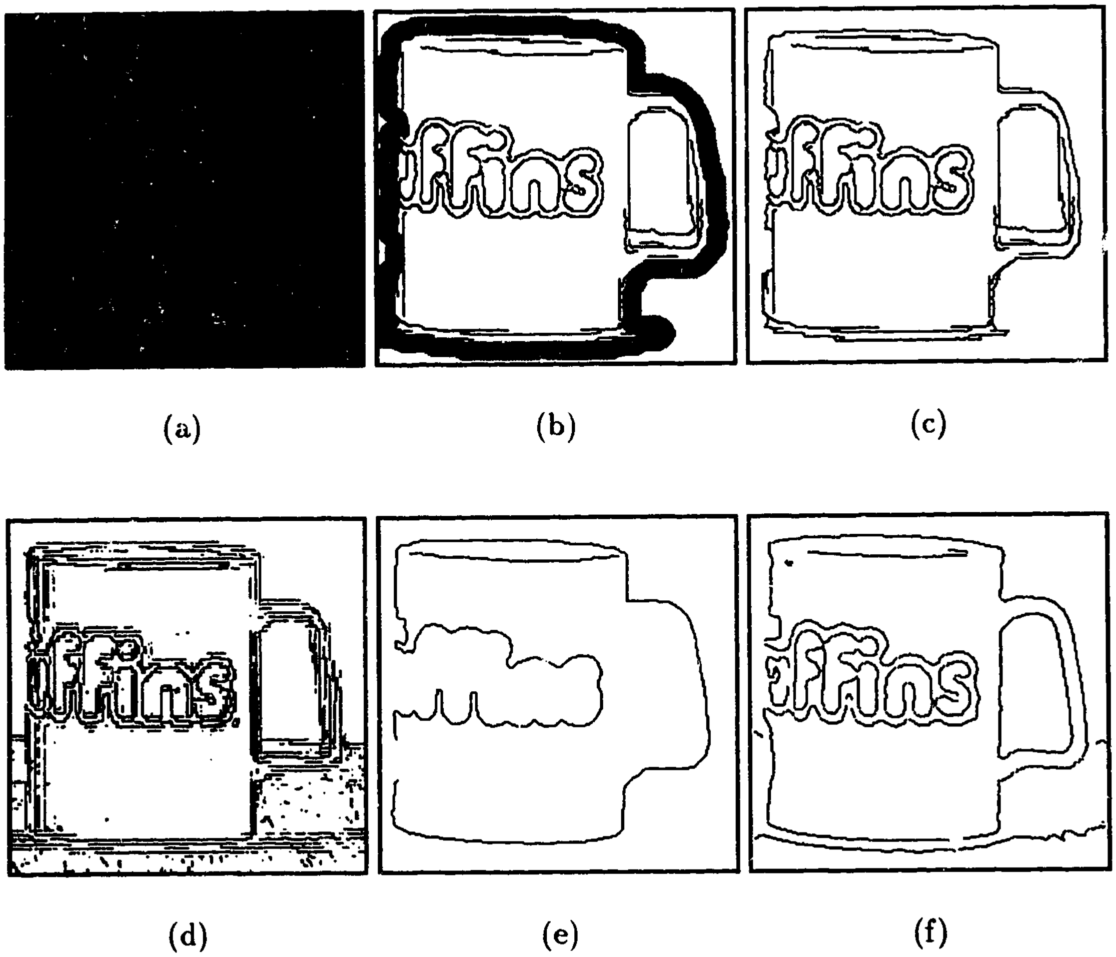


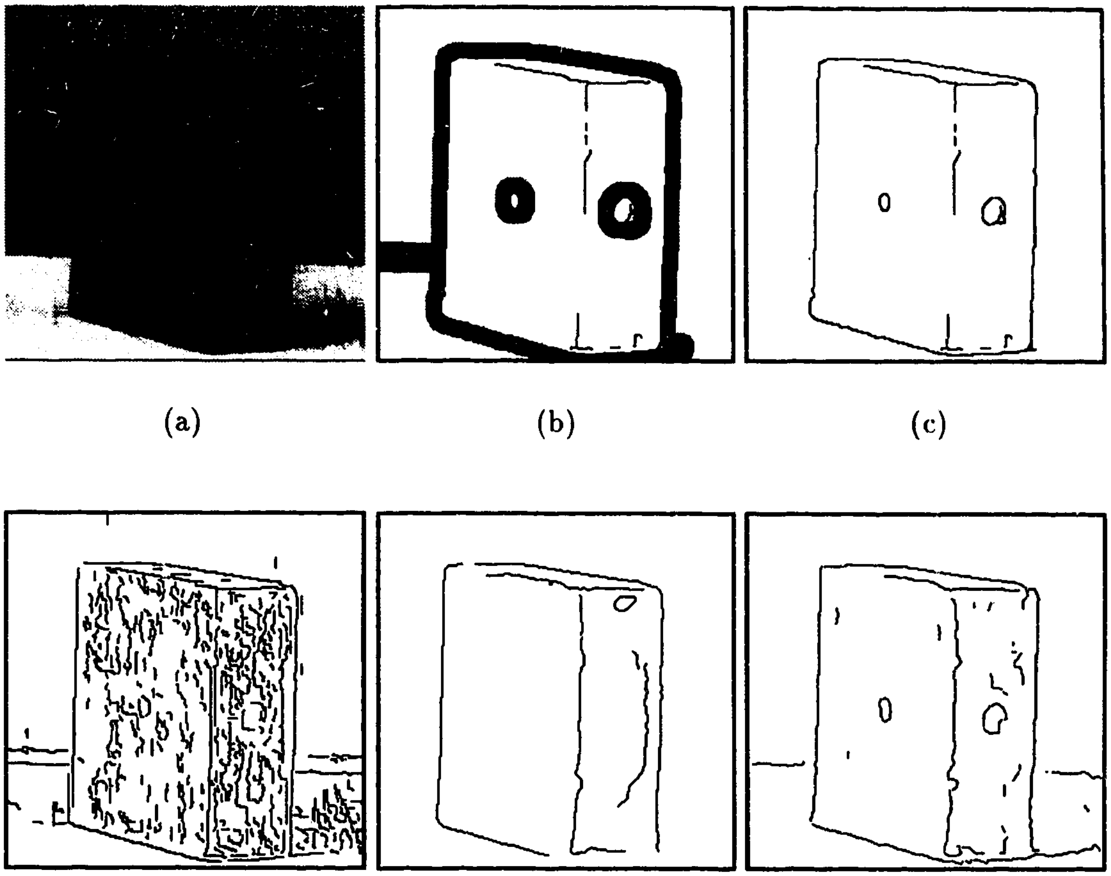
Figure 4.10: Boundary detection result on a test image with three objects



(a) (b) (c)

(d) (e) (f)

Figure 4.11: Boundary detection results of the four methods on a big cup image



(a)

(b)

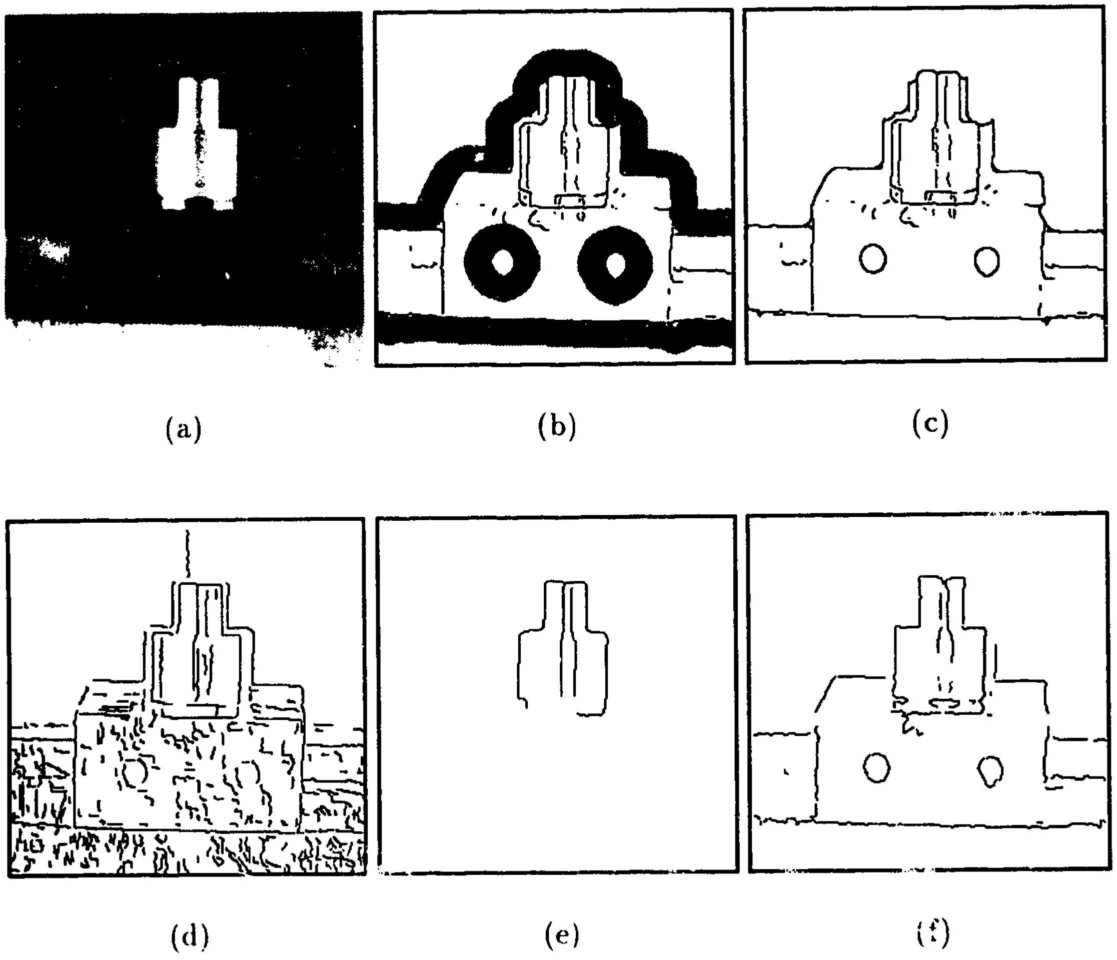
(c)

(c)

(d)

(e)

Figure 4.12: Boundary detection results of the four methods on a block image



(a) (b) (c) (d) (e) (f)

Figure 4.13: Boundary detection results of the four methods on an image with two parts

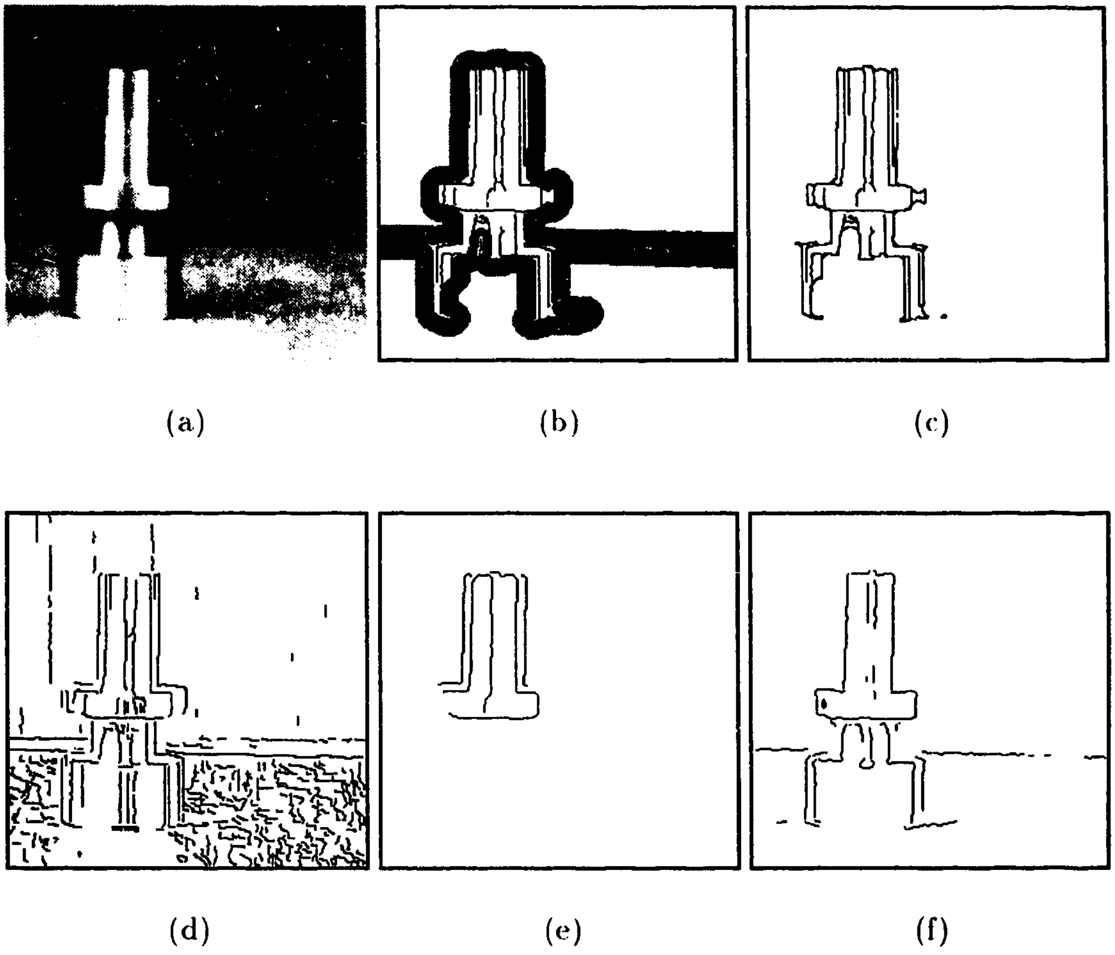
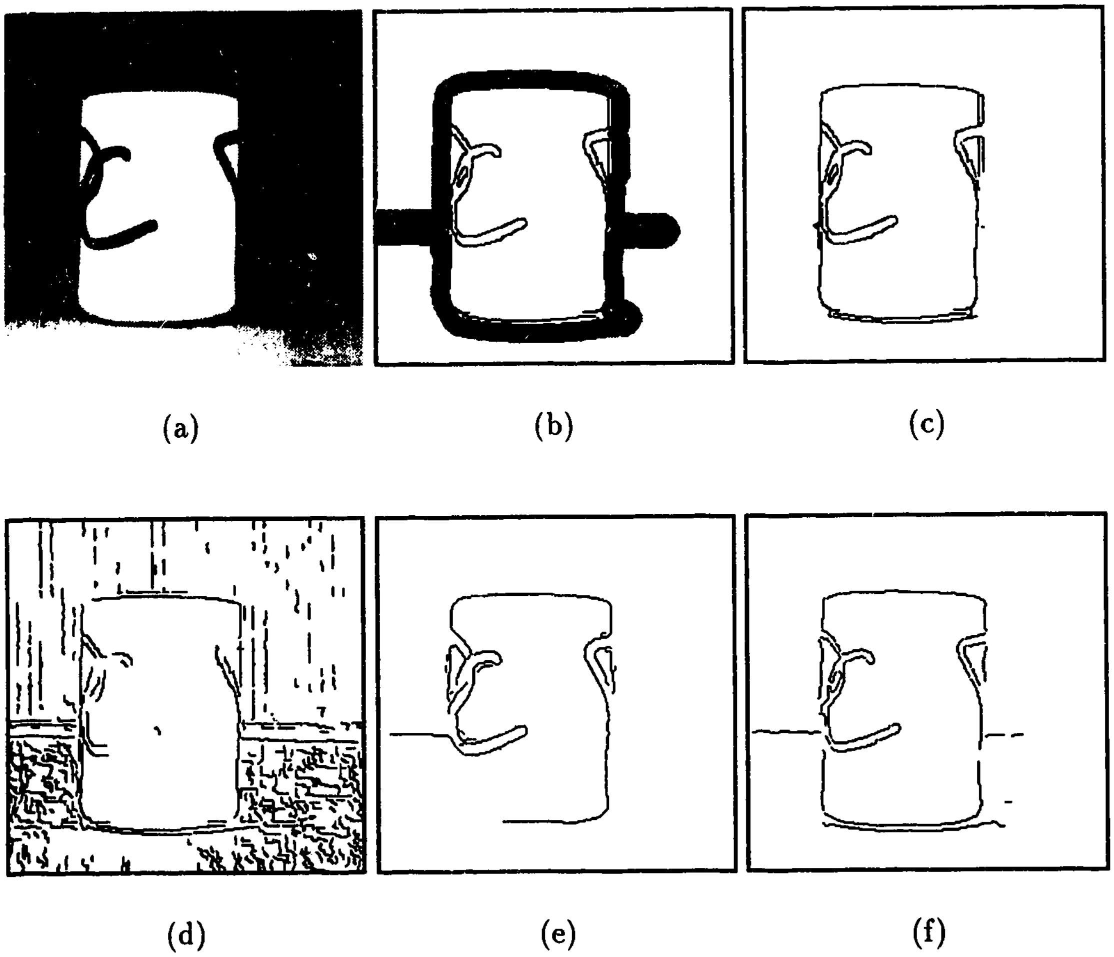


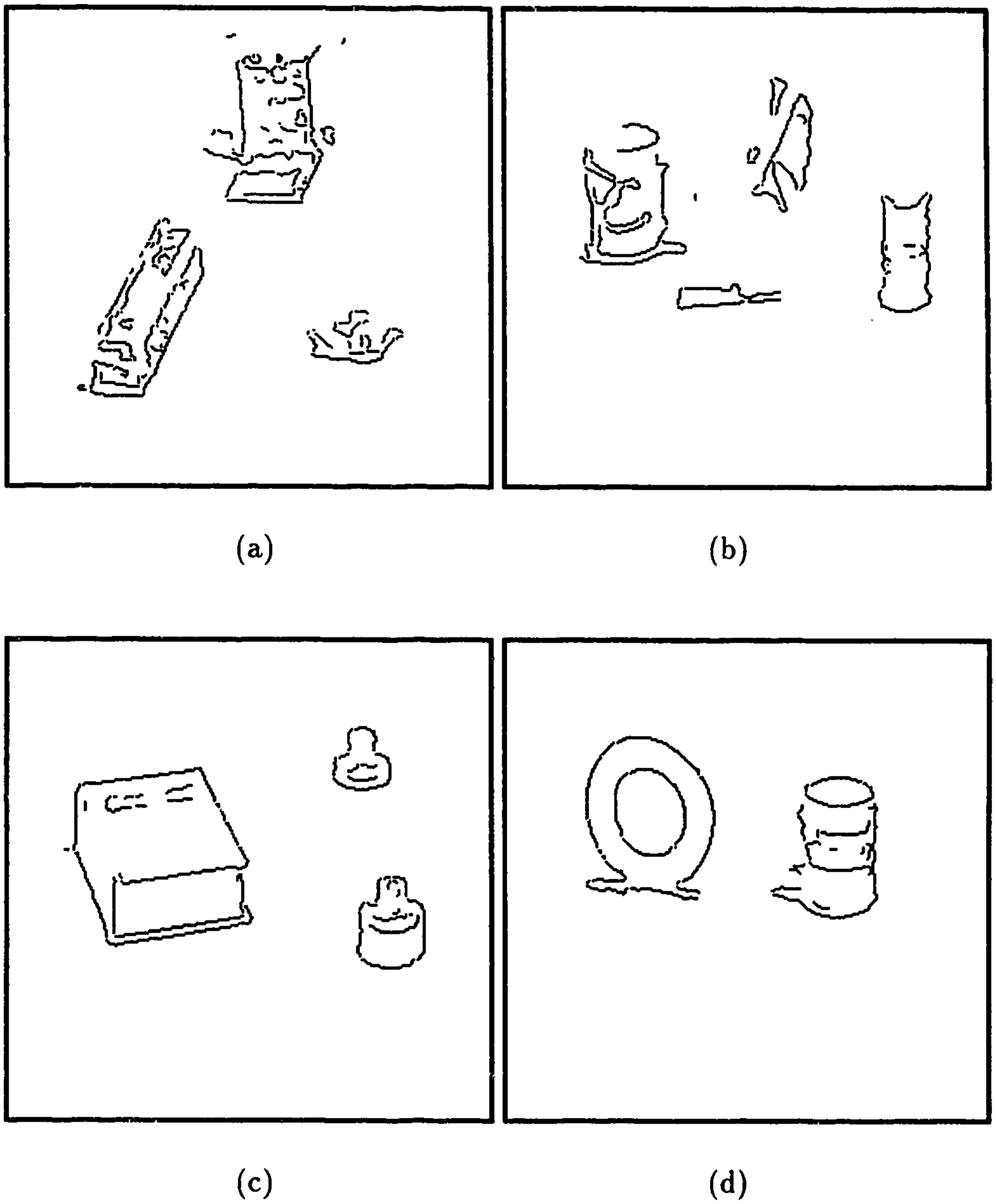
Figure 4.14: Boundary detection results for the four methods on a machine part image



(a) (b) (c)

(d) (e) (f)

Figure 4.15: Boundary detection results of the four methods on a small cup image



(a)

(b)

(c)

(d)

Figure 4 [illegible] Results of applying the VR method to four images

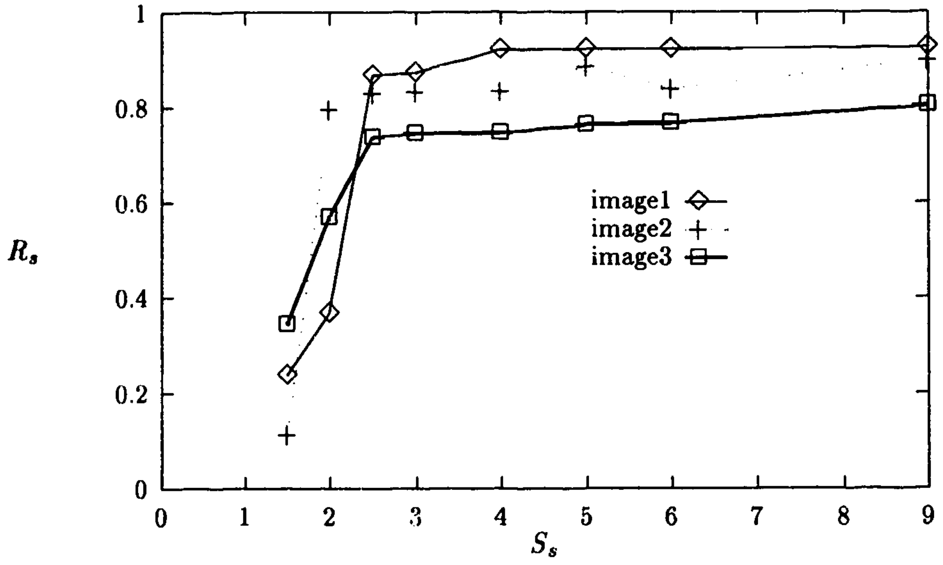


Figure 4.17: Relation between R_s and S_s for three test images

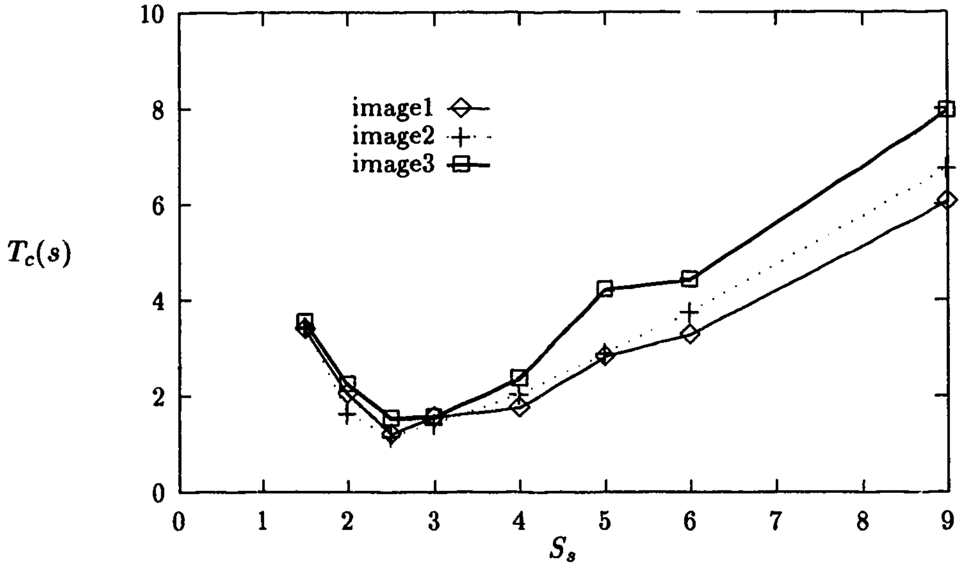


Figure 4.18: Relation between C_i and S_s of the three test images

Chapter 5

Moving Object Detection Using Dynamic Dilations

Moving object detection is a basis for motion analysis. In this chapter, two algorithms based on dynamic dilations are proposed and implemented to detect moving objects from an image sequence. Roll dilation and expand dilation are employed respectively. These methods can detect one or multiple moving objects. In the algorithms, dynamic dilations are used to extract moving objects as connected regions. Motion analysis can be carried out on the shape of each detected moving object. Each method has been implemented on two image frames.

5.1 Previous Work

Moving object detection is one of the important topics in image processing and has been studied extensively [14] [24] [44] [47] [32]. In some applications it is necessary to detect motion of more than one object In the following discussions the focus will be on the cases in which there are two or more moving objects.

A tracking algorithm based on binary object forest was introduced by Nicole and Fiebig [33]. In this method, an input image is sliced into a series of binary images using a set of thresholds. For example, if there is an image **F** and a set of thresholds t_k (k = 1,2,..., n), then **F** can be decomposed into a set of binary images $\mathbf{F}_k$. The pixels in $\mathbf{F}_k$ are set to 1 if the corresponding pixel in **F** is greater than t_k, otherwise it will be set to 0. In the binary image set, each connected region is defined to be an *atom*. Information such as size and centroid of each atom is computed and recorded. Thus the structure of an object is represented by trees between atoms within a binary image and trees of atoms between adjacent slices. Motion is detected by matching tree representations of different objects in different image frames. This method is efficient for images in which each component of an object has a different gray level. If an object has a smooth gradient the method will not work very well.

Bergen et al. [6] proposed an algorithm to estimate motions of two moving patterns within the neighborhood of an image point. The algorithm uses three image frames and does not require segmentation. In this iterative method the motion of one component is estimated first. It is then removed from the image so that estimation of the second component can be performed more precisely. This algorithm imposes two restrictions on the image sequence. First, the time interval between two successive image frames must be very small so that the acceleration of each moving component can be ignored. Secondly, the method can only be applied to two moving components.

A method based on object models was proposed by Huttenlocher et al. [18]. In this algorithm the image of a moving object is decomposed into a 2-D shape change and a 2-D motion. Then 2-D geometric models are used to capture the shape changes of each moving object between successive frames. These models provide global geometric constraints for tracking an object. The 2-D models are matched using a distance measurement to estimate the motion. An example of an image sequence of two moving objects was given. It was not made clear that how the method can be applied to more moving objects.

In the above methods, image sequences are taken from static cameras. Optical flow is a popular tool used for motion analysis. There are also algorithms for motion detection using active cameras. Optical flow methods use different techniques to detect the motion field in an image sequence from the variations of the image intensity, and assign a velocity to each pixel of the image sequence. Differential techniques compute velocity from spatiotemporal derivatives of the intensity of an image or filtered versions of the image [16]. Region-based matching approaches define the velocity as the position shift that produces the best fit between image regions at different times [1]. A third class of optical flow methods is based on the output energy of velocity-tuned filters which are defined in the Fourier domain [4]. Another technique is based on the behavior of band-pass filter outputs [12]. Optical flow methods can handle multiple moving objects but usually the motion of each object between two successive image frames is restricted to several pixels. Barron et al. [5] implemented and compared some existing optical flow algorithms.

Murry et al. [30] proposed an algorithm to track a moving object using an active camera. In their work, the pin-hole camera model is used. The camera is installed on a pan/tilt equipment and can rotate along two axes. By building up the relation between each pixel's position in two image frames taken at different time, the background motion which is caused by the camera rotation is compensated by mapping the pixels that correspond to the same real scene point to the same image plane. The image subtraction technique is applied to the two images to get the moving regions. Using this technology the first image is subtracted from the second image at each pixel and the absolute value is calculated and thresholded by an appropriate value. Since the subtraction technique has the drawback that the center of the detected moving region is about the average of the actual positions of the object in the two images, moving edges are used to represent the moving object. Two methods are employed to get moving edges. The first is to perform a logical AND of the thresholded difference image and the image of each frame. The second method is to multiply the strength of the edge image with the difference image followed by a thresholding operation. Due to the errors in the pan/tilt angles, there are false motions in the results. The authors employed conventional morphological operations to remove the false motions from the thresholded difference image. A morphologicai erosion is first applied to delete small moving regions and then a dilation is used to recover the real moving object. In a work by Elnagar et al. [13], the camera can not only pan/tilt but also translate in the cene. They developed a series of rotation and translation equations to build up a corresponding relation of pixels in two continuous image frames. The success of the algorithm is based on the theory that if a pixel comes from a point on a moving object in the scene, it dose not satisfy the planarity constraint. On the other hand a pixel related to a static point in the scene will satisfy the constraint. In this method traditional mathematical morphology is also applied to the detected motion fields to remove false motions caused by the equipment. An erosion is used first to remove the narrow regions, then a dilation is applied to the eroded result to restore the wider regions. The experimental results of detecting two moving objects between continuous image frames are presented.

Many existing algorithms for static cameras only process image sequences with one or two moving objects. In the methods which handle more than two motions, the objects are usually in simple shapes. Other restrictions such as limited moving distance are imposed by some algorithms. In this chapter two algorithms based on dynamic dilations are proposed to detect multiple moving objects. These methods are different from the optical flow methods. They extract the shapes of moving objects and each object is detected as a whole. Motion analysis can be performed by matching these detected moving objects.

5.2 Problem Formulation

The following points list the properties of image sequences and moving objects that can be processed using the proposed methods.

- The input image sequence is taken by a static camera.
- The sequence may contain more than two moving objects.
- There is no restriction on the size of an object and the distance this object can move.
- The moving objects are allowed to disappear from or appear in one image frame in some cases.
- The background will not change greatly from frame to frame.
- Objects are visually different from the background and their shapes do not change greatly while moving. They can rotate along the axis perpendicular to the image plane.

5.3 The Algorithm Based on Roll Dilation

The first method is developed based on roll dilation. In the following discussions only two image frames are used to illustrate the structure of the algorithm. This algorithm can be divided into four phases: preprocessing, moving edge detection, object detection and matching. Each one will be discussed in detail. Figure 5.7 is a general overview of the algorithm.

5.3.1 Algorithm Description

Preprocessing

Suppose two successive input images $\mathbf{F}_1$ and $\mathbf{F}_2$ are both of size $M \times N$. A median filter is used to remove the possible random noise in them. The difference image $\mathbf{F}$ is then generated from the two filtered images using

$$f(i,j) = |(f_1(i,j) - f_2(i,j))| \quad (i = 1,2,\dots,M, \quad j = 1,2,\dots,N).$$

Since gray values of the background remain constant or only change slightly from $\mathbf{F}_1$ to $\mathbf{F}_2$, $\mathbf{F}$ should only contain the changing parts of the image sequence. $\mathbf{F}$ is thresholded to a binary image $\mathbf{B}$ with a threshold T_b:

$$b(i,j) = \begin{cases} 255 \text{ (white)} & \text{if } f(i,j) > T_b, \\ 0 \text{ (black)} & \text{otherwise.} \end{cases}$$

$\mathbf{B}$ is then dilated by a suitable structuring element. After the dilation it should contain several connected regions which correspond to the possible moving objects. Edge detection is applied to $\mathbf{F}_1$ and $\mathbf{F}_2$ generating edge maps $\mathbf{E}_1$ and $\mathbf{E}_2$. After edge detection $\mathbf{E}_1$ and $\mathbf{E}_2$ are filtered to remove noise edges. This is done by applying a strength thresholding. The regions in $\mathbf{B}$ are used to mark the edges in $\mathbf{E}_1$ and $\mathbf{E}_2$. Corresponding to each white pixel in $\mathbf{B}$, if the pixel at the same position in $\mathbf{E}_1$ or $\mathbf{E}_2$ is an edge pixel it is marked. The unmarked edges are then removed.

Moving Edge Detection

After preprocessing, an edge map which contains edges of possible moving objects is generated for each frame. An Exclusive-Or operation of the two edge maps is then conducted to obtain the changed edges in each edge image. The resulting images are $\mathbf{E}_1'$ and $\mathbf{E}_2'$,

$$\mathbf{E}_1' = \{\mathbf{e} \mid \mathbf{e} \in \mathbf{E}_1 \text{ and } \mathbf{e} \notin \mathbf{W}^{\mathbf{e}}_{\mathbf{E}_2}\}$$

and

$$\mathbf{E}_2' = \{\mathbf{e} \mid \mathbf{e} \in \mathbf{E}_2 \text{ and } \mathbf{e} \notin \mathbf{W}^{\mathbf{e}}_{\mathbf{E}_1}\}.$$

$\mathbf{W}^{\mathbf{e}}_{\mathbf{E}_2}$ and $\mathbf{W}^{\mathbf{e}}_{\mathbf{E}_1}$ are small neighborhoods of an edge pixel $\mathbf{e}$ and

$$\mathbf{W}^{\mathbf{e}}_{\mathbf{E}_1} = (\mathbf{W} + \mathbf{e}) \cap \mathbf{E}_1, \quad \mathbf{W}^{\mathbf{e}}_{\mathbf{E}_2} = (\mathbf{W} + \mathbf{e}) \cap \mathbf{E}_2$$

in which $\mathbf{W}$ is a small window. $\mathbf{W}^{\mathbf{e}}_{\mathbf{E}_1}$ and $\mathbf{W}^{\mathbf{e}}_{\mathbf{E}_2}$ are used to adjust the possible position shift of an edge pixel in the two images. The shift may be caused by noise or illumination change on the background and the motionless objects. The size of $\mathbf{W}$, $m \times n$, depends on the edge quality. After the Exclusive-Or operation, only those edge segments that belong to the moving objects in $\mathbf{F}_1$ are kept in $\mathbf{E}'_1$. As well $\mathbf{E}'_2$ should contain the edge segments of moving objects in $\mathbf{F}_2$. The rest of the procedure is then similar to that of the boundary detection algorithm introduced in Chapter 4. A dilation is applied to the edges in $\mathbf{E}'_1$ and $\mathbf{E}'_2$ to group the moving edges into connected regions. The edges in each region is defined to be a group which corresponds to a potential moving object.

Moving Object Detection

Now roll dilation can be applied to the edge groups in $\mathbf{E}'_1$ and $\mathbf{E}'_2$ to detect the moving objects. The detail of this step is left out since it is similar to that in the boundary detection method. Please refer to previous chapter for the detail.

Matching

Shape features of each detected moving object are calculated and compared with those of the objects in the other image frame. The two objects from different frames with the closest shape features are considered to be the same object. Non matching objects are considered to have either entered into or exited from the image or are currently occluded. Matching can be carried out in a multi-dimension feature space. Feature vector $M = (m_1, m_2, \ldots, m_k)^T$ of each detected object is used to determine the distance between any two objects from two different frames in the feature space. Object pairs with the shortest distance are considered to match each other.

Shape signature is a weighted combination of different object shape features. Suppose an object has k features which are m_1, m_2, $\ldots$ and m_k and the corresponding weights are w_1, w_2, $\ldots$ and w_k. The shape signature of the object is defined as

$$SS = \sum_{l=1}^{k} w_l m_l.$$

For each object i in the first image, shape signature SS_{1i} is calculated. Compare it with the signature SS_{2j} of object j in the other image, the deviation is calculated as

$$Dev_{1i2j} = |\frac{SS_{1i} - SS_{2j}}{SS_{1i}}|.$$

The matching result is

$$Match_i = j \text{ with the minimum } Dev_{1i2j}.$$

Currently the size and some simple moments of an object are used as its shape features and they produced good results in some experiments.

This matching method is based on simple shape features of the detected moving objects. Its accuracy can be enhanced by detecting the constituent parts of each object and using high level image understanding techniques in the matching process. For example, a moving object can be represented by a graph in which each node represents a component of the object. Then the graph representation of each object can be used for matching. The morphological signature proposed by Loncaric et al. [25] can also be used to improve the matching result. The basic idea is to define a set of different structuring elements and erode an object with the structuring element set iteratively at different rotation angles. The iteration number used by each structuring element to eliminate the object at different angles is recorded as the morphological signature of the object.

When an object is occluded partially, shape features of the object will change greatly and the matching will be more difficult. One possible solution is to use structural description of the object. If the object moves out the image or occluded totally by another object or if an object moves into the frame or comes out from the occlusion of another object, it can be distinguished if there is no other moving object that can match.

Pseudo Code

Following is the pseudo code of the whole algorithm.

```
PHASE 1: preprocessing
    INPUT image F1, F2
    FILTER F1, F2 to remove random noise
    DETECT edges of F1, F2
        edge image 1 E1 = edge map of F1
        edge image 2 E2 = edge map of F2
    REMOVE noise edges in E1, E2
    THRESHOLD abs|F1 - F2| to get B
    DILATE B into several regions
    MARK E1 and E2 with regions in B
PHASE 2: moving edge detection
    XOR E1 and E2 to get the moving edges
    DILATE moving edges into groups
PHASE 3: moving object detection
    FOR(;;)
        IF there is no group in E1 and E2
            GOTO PHASE 3
        ELSE
            PICK one group from E1 or E2
            MARK the edges in the group
            ROLL DILATE outside the moving edges
        ENDIF
    ENDFOR
PHASE 4: matching
    CALCULATE shape features of each moving object
    FOR each object in frame 1
        COMPARE its features with that of objects in frame 2
        SELECT the one with the shortest distance as the matching object
        REMOVE matched objects from both frames
    ENDFOR
```

5.3.2 Experimental Results

The algorithm has been tested on different image sequences and the results are encouraging. These test images contain different moving objects. The analysis of these results is similar to that in Chapter 4. The experimental results of four image sequences, shown in Figure 5.2 through Figure 5.5, are summarized. The images in Figure 5.4 and Figure 5.5 are courtesy of John Barron. In his work [5] these images are used to evaluate the performance of different optical flow methods. The figures are all arranged in the same way. Parts (a) and (b) are the original images. The edge images are shown in (c) and (d) corresponding to the originals. The detected moving edges are shown in (e) and (f) correspondingly. The results of applying the algorithm are shown in (g) and (h) where each detected moving object is surrounded by the trace of the structuring element in the roll dilation.

Figure 5.2 gives an example in which there are two moving objects of similar sizes in a simple background. The algorithm has detected the two moving objects correctly. Figure 5.3 shows another example in which there are two moving objects. One is a wood toy which was moved and the other is a desk drawer which was closed. The shape of the wood toy has been deformed as it has similar gray value with its neighbor but the two objects have still been detected. In Figure 5.4 there is a Rubik Cube which turns on a platform. Although it is difficult for this algorithm to detect slightly turning objects, the cube and the platform have been detected thanks to the rich inside edges. Figure 5.5 shows another example with four moving objects, a man, a taxi, a car and a van. The van at the right margin of the image and the car on the left side have been detected with deformed shapes since they have similar gray values with the background. Shapes of the other two moving objects have been detected correctly.

5.4 The Algorithm Based on Expand Dilation

The method based on roll dilation uses moving edges to detect objects. If an object has straight boundaries and it moves along one of them, the overlapped boundary in the two image frames will be missing after moving edge detection. If the object surface is smooth so that inside edges are scarce, the method will fail. Figure 5.6 shows an example in which a moving object cannot be detected successfully since its overlapped boundaries have been removed in the detected moving edges. In such cases expand dilation can be used to detect the object shape from inside an object. Following is the moving object detection algorithm using expand dilation.

5.4.1 Algorithm Description

This algorithm can also be divided into four phases: preprocessing, morphological object expansion, morphological merging and matching. Following is a detailed description of each step. Figure 5.7 illustrates a general overview of the algorithm when used to process two image frames. In a real application more frames can be employed to get a more accurate motion estimation.

Preprocessing

This step is quite similar to that of the method based roll dilation. A median filter is used to remove random noise in $\mathbf{F}_1$ and $\mathbf{F}_2$. A difference image **B** is then generated by subtracting the two images and thresholding the absolute value of the result at each pixel. The role of **B** is however different from that in the previous method. Pixels in **B** will be used as seed points for expand dilation. Edge detection is applied to $\mathbf{F}_1$ and $\mathbf{F}_2$ generating edge maps $\mathbf{E}_1$ and $\mathbf{E}_2$ which are then filtered to remove noise edges.These edge images will be involved in the condition of the expand dilation.

Morphological Object Expansion

It is reasonable to say that in most images, objects can be distinguished from the background by considering the difference in average gray level. In other words, there is usually an sharp gray value change from an object to the background and this is the basis of many edge detection methods. Based on this knowledge and a seed point (i_s, j_s) in **B**, expand dilation is used to extract the shape of a moving object. During the preprocessing stage, background is removed from **B** because it has similar gray values in the two images. The white pixels in **B** indicate the regions which have changed from one image to the next in the sequence so that each white pixel can be considered as a seed point.

There are three parts of the condition C for an expand dilation process, the first two are based on edge images and the third one is defined on the average gray value deviation within a neighborhood covered by a structuring element on image $\mathbf{F}_1$ or $\mathbf{F}_2$. Using the first two parts of the condition is usually enough to produce a good result. The last sub-condition is also employed here to get a better output. After each expand dilation process, a connected region in an edge image has been formed. This region may be an object or only a component of the object. It can also be a part of the background. An object is usually composed of different facets or different components. Correspondingly in the edge image of the object there are edges inside the object and the object shape is thus divided into several regions. The expand dilation process can be repeated to find out each part of the object. The result of each dilation is marked and in a later dilation process, pixels that fall into these marked regions will not be considered as seed points.

Since the difference image corresponds to all changed regions in the image, it may happen that in one frame a seed point is inside an object, and in the other frame the same point falls into the background or the inside of an object which is occluded by the object in the first frame. In the first case if the background is clean the expansion result will fill the whole background and a threshold on the size of the result can tell if it is a valid region of an object. In the second case, the occluded object will be considered as one moving object and it can be eliminated in the matching process if its shape is different from other moving objects.

In summary the object expansion process can be described as:

1. Find a white pixel (i_s, j_s) in the difference image $\mathbf{B}$. Its position is used as a seed point.

2. In $\mathbf{E}_1$ or $\mathbf{E}_2$, if (i_s, j_s) has not been marked as *searched* or *object*, expand from point (i_s, j_s) using expand dilation to include all positions that satisfy condition C.

3. If the formed region is classified as an object component, mark it *object,* otherwise mark it *searched.*
4. Change $b(i_s, j_s)$ to black.
5. If there is any white pixel left in **B**, goto 1, otherwise stop.

It is obvious that object complexity will affect the expansion result. In this algorithm, interior edges are used to represent the complexity of an object. The more complex the object, the more edges there will be within the object.

Morphological Object Merging

Using expand dilation, some regions which are components of moving objects have been detected in the images. The edges inside an object usually divide the object into several regions which need to be merged. In the merging process, morphological closing is used.

Matching

The matching process is the same as that of the roll dilation based method.

Pseudo Code

Following is the pseudo code of the algorithm.

```
PHASE 1: preprocessing
    FILTER input image F1 and F2
    THRESHOLD $abs|F1-F2|$ to get B
PHASE 2: morphological expansion
    FOR(;;)
        IF there is no white pixel in B
            GOTO PHASE 3
        ELSE
            PICK one white pixel $(i_s, j_s)$ from B
            IF $(i_s, j_s)$ is not marked in E1 or E2
            EXPAND DILATE from $(i_s, j_s)$ in E1 or E2
                IF the formed region falls between preset thresholds
                    MARK it as an "object" region
                ELSE
                    MARK it as a "searched" region
                ENDIF
            ENDIF
            CHANGE $b(i_s, j_s)$ to black
        ENDIF
    ENDFOR
PHASE 3: morphological merging
```

```
    DILATE all regions marked as "object" in E1, E2
    ERODE the merged object to its original shape
PHASE 4: matching
    CALCULATE shape features of each object in E1, E2
    FOR each object in E1
        COMPARE its features with that of objects in E2
        SELECT the one with the shortest distance as the matching object
        REMOVE this object from E2
    ENDFOR
```

5.4.2 Experimental Results

The proposed algorithm has been tested on different image pairs with encouraging results. Each test image contains at least two moving objects and its complexity varies in terms of object and background. In the following experimental results from four image pairs, shown in Figures 5.8 through Figures 5.11, are summarized. These figures are all arranged in the same way. In each figure (a) and (b) show the original images, (c) is the thresholded difference image. Edge images corresponding to the originals are shown in (d) and (e). The results are shown in (f), (g) and (h). In (f) and (g) each detected moving object is marked in dark gray with a white boundary which is the trace left by merging different neighbor regions. The background that has been searched is marked in gray. Part (h) shows the estimated motion of each detected moving object with an arrow illustrating both the direction and the distance. Figure 5.8 shows an example with three moving objects of different sizes. The book moved to the right and rotated slightly, the file holder moved to the left and the camera turned slightly. All their shapes have been detected with deformation of different degrees and this has affected the estimated motion slightly. The examples shown in Figure 5.9 and Figure 5.10 use the the same original images as in Figure 5.2 and Figure 5.3. The algorithm has detected the motion of each object correctly. Figure 5.11 is an example with two big moving objects, a chair and a monitor. Both objects have been detected but the chair in the second image has been connected to some regions with the background.

5.4.3 Result Analysis

The performance of this algorithm is dependent on the shape of each detected moving object. Three factors affecting this are the edge map quality, the average gray value deviation threshold and the size of the structuring element used in the expansion process. In this discussion the relation between the experimental results and these three factors will be explored. In further experiments these factors can be set according to this relationship so that better results can be obtained.

In addition to the previously defined success rate R_s and structuring element size S_s, two more definitions are given to quantitatively describe the above mentioned factors. In an edge image, *edge density* is defined as

$$D_e = \frac{N_e}{N_t} \times 1000$$

in which N_e is the number of true edge pixels and N_t is the total number of image pixels. Suppose N_n is the number of noise edge pixels, *noise edge rate* is defined as

$$R_n = \frac{N_n}{N_e} \times 100.$$

In real applications N_n can be approximated by the number of noise edge pixels that have been removed in the preprocessing stage. Since discs work better than other shapes when an object shape is unknown, they are used as structuring elements in the experiments reported here. Figure 5.12 shows the relation between R_s and D_e using different structuring elements for the test image sequence in Figure 5.8. To see clearly the effect of D_e and S_s on R_s, T_d is fixed. For each structuring element, there is an edge density range within which the algorithm has the best success rate. Other tested images give similar results. Figure 5.13 shows the relation between R_s and T_d using different structuring elements on the same image pair. From these curves we can see that there is also a range of T_d within which the success rate of the algorithm is fairly good. For other image sequences, the same conclusion can be reached. Figure 5.14 shows the effect of S_s on R_s for different test images. From the figure it can be seen that when S_s falls between 1 and 3 pixels, the success rate is the highest. For different images and different edge detection methods this best structuring element range could be different. The above analysis implies that for each of the three factors, i.e. edge quality, average gray value deviation threshold and structuring element size, there is a range within which the algorithm produces the best results on most test images. This best range for edge density is about 150 to 200 for small structuring elements and about 80 to 180 for large ones. Setting T_d between 5 and 10 and S_s to 1, 2 or 3 pixels produced good results for the tested image pairs.

In the worst case the number of operations used to detect the moving objects is $M \times N \times K$ where $M \times N$ is the size of the image and K is the structuring element size. To make it simple, let $M = N$. For a disc structuring element, K can be expressed as $K = 2\pi S_s^2$. The number of the operations will be $2\pi N^2 S_s^2$. So the complexity of the algorithm is $O(N^2 S_s^2)$.

5.4.4 Possible Improvements

Threshold T_b used for producing the difference image **B** affects the number of seed points. Decreasing T_b increases the number of seed points and vice versa. Sometimes more seed points can avoid the case that an expansion process is stopped by an interior edge, because the expansion process can restart from a seed point on the other side of the edge. Figure 5.15 is an example in which one more seed point has improved the result. Motionless parts in **B** have gray values near *0*, and moving parts correspond to gray values much greater than *0*. This means the selection of T_b can use the histogram of the difference image. It is also possible that a feedback system be built to adjust the threshold in case it fails to detect any moving objects. The value of T_d should be set so as to get a balance between the expansion process stopping too soon and leaking out of an object. The previous experimental data can be used to set an initial value. It is not easy to determine T_d automatically as we do not know exactly where the greatest gray value changing is located. One potential method is to calculate T_d along edges because there are obvious gray value changes in these places. In a gray value image, D_g is calculated and summed along edges and then it is averaged to obtain T_d. Finally T_d is decreased by a small fraction to adjust it for the average gray value deviation at edge gaps. Size and shape of the used structuring element play an important role in the algorithm. It is already known that discs do better than other ones and those ones with radius ranging from 1 to 3 pixels have produced good results for the tested images. One way to define S_s automatically is to calculate the size of each gap in the edge map. Another way is to build a feedback system so that when the system fails to detect moving objects in an image, the size of the structuring element is increased.

5.5 Summary

In this chapter two algorithms are designed and implemented to detect moving objects in an image sequence. These algorithms are based on the newly defined dynamic dilations which are powerful in detecting shapes of objects. The detected shape of each moving object can be used to estimate its motion in the image sequence. These two algorithms complement each other. The roll dilation based method can be used in cases when there is no boundary overlap between two frames, or each object has many inside edges. On the other hand, the expand dilation based method can deal with the case with boundary overlap and simple object surface.

Compared with other methods, following are some characteristics of the proposed methods:

- Using dynamic dilations, the implementation of each algorithm is simple.
- These algorithms can detect one or multiple moving objects in an image sequence.
- They can process scenes with objects of various shapes and sizes as well as images with complex backgrounds.
- The methods can detect slight motions as well as large movements of the objects between different frames.
- The algorithms are relatively robust within certain noise level and edge quality.

In the above discussions only two image frames are used. By employing more image frames, these methods can produce a more accurate estimation of each moving object. Figure 5.16 shows a structure for applying expand dilation based method to more frames. In this figure $\mathbf{F}_1$, $\mathbf{F}_2$, ..., $\mathbf{F}_n$ are successive image frames, $\mathbf{E}_1$, $\mathbf{E}_2$, ..., $\mathbf{E}_n$ are their edge images, $\mathbf{B}_1$, $\mathbf{B}_2$, ..., $\mathbf{B}_{n-1}$ are difference images and M_1, M_2, ..., M_{n-1} are detected motions. With multiple image frames, a previous difference image is used to compensate the present one so that more seed points can be located to improve the expansion result. Each edge image is used in generating the next one with more precision. With each new image frame, the algorithm will estimate the motion with more accuracy. In Figure 5.16, M_{n-1} is the motion of the whole image sequence.

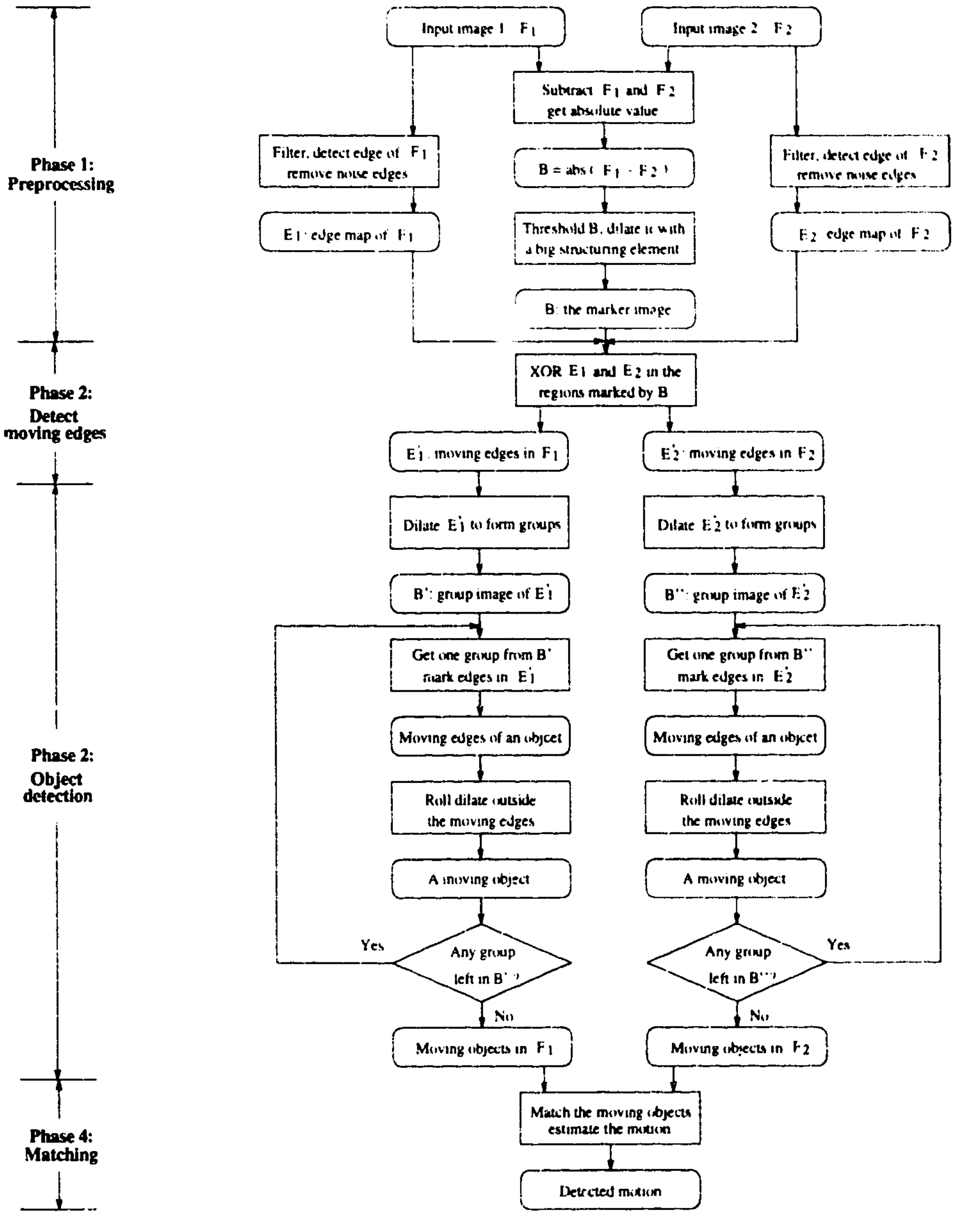


Figure 5.1: Structure of the method based on roll dilation

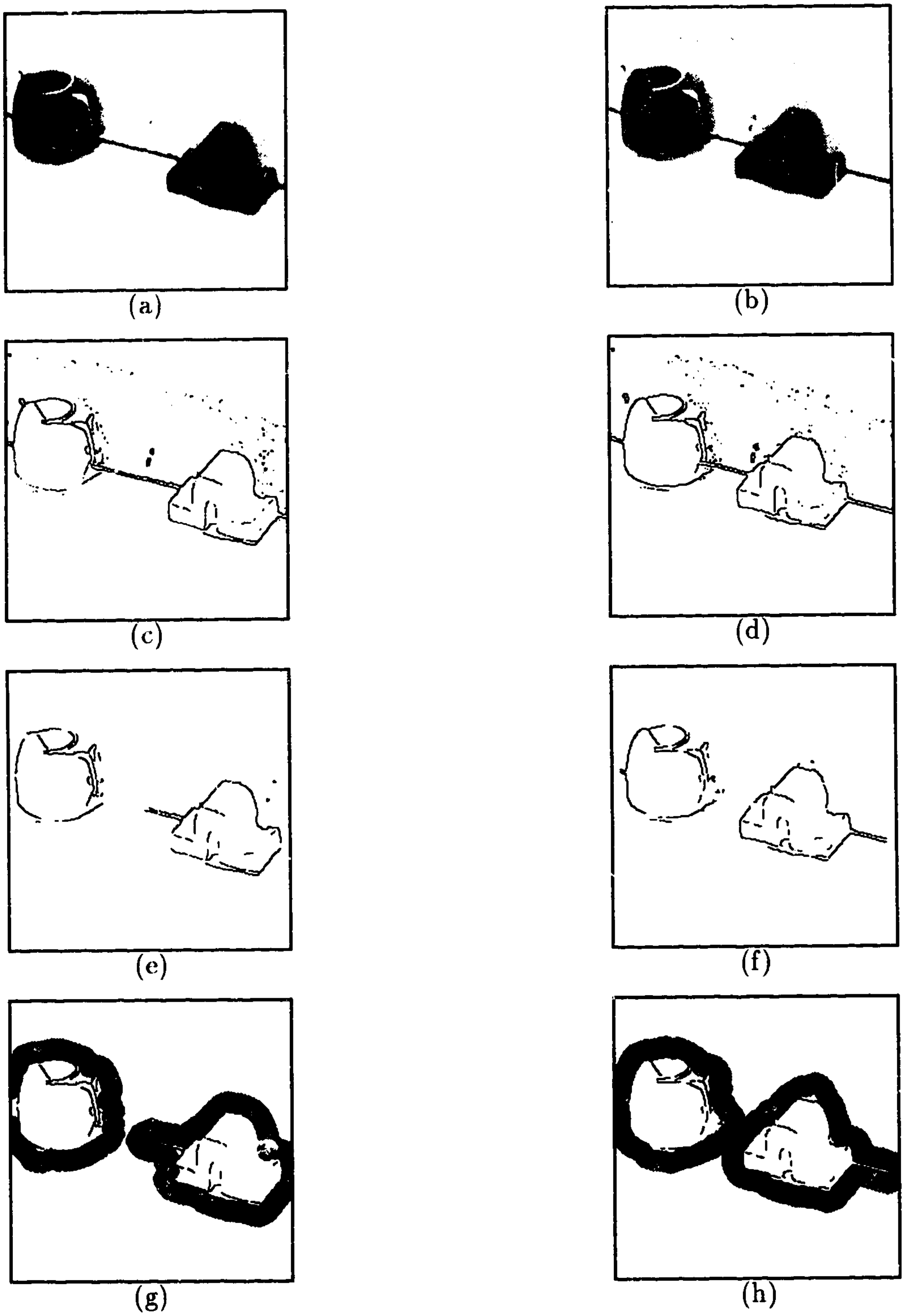


Figure 5.2: Results of applying the first moving object detection method to an image pair

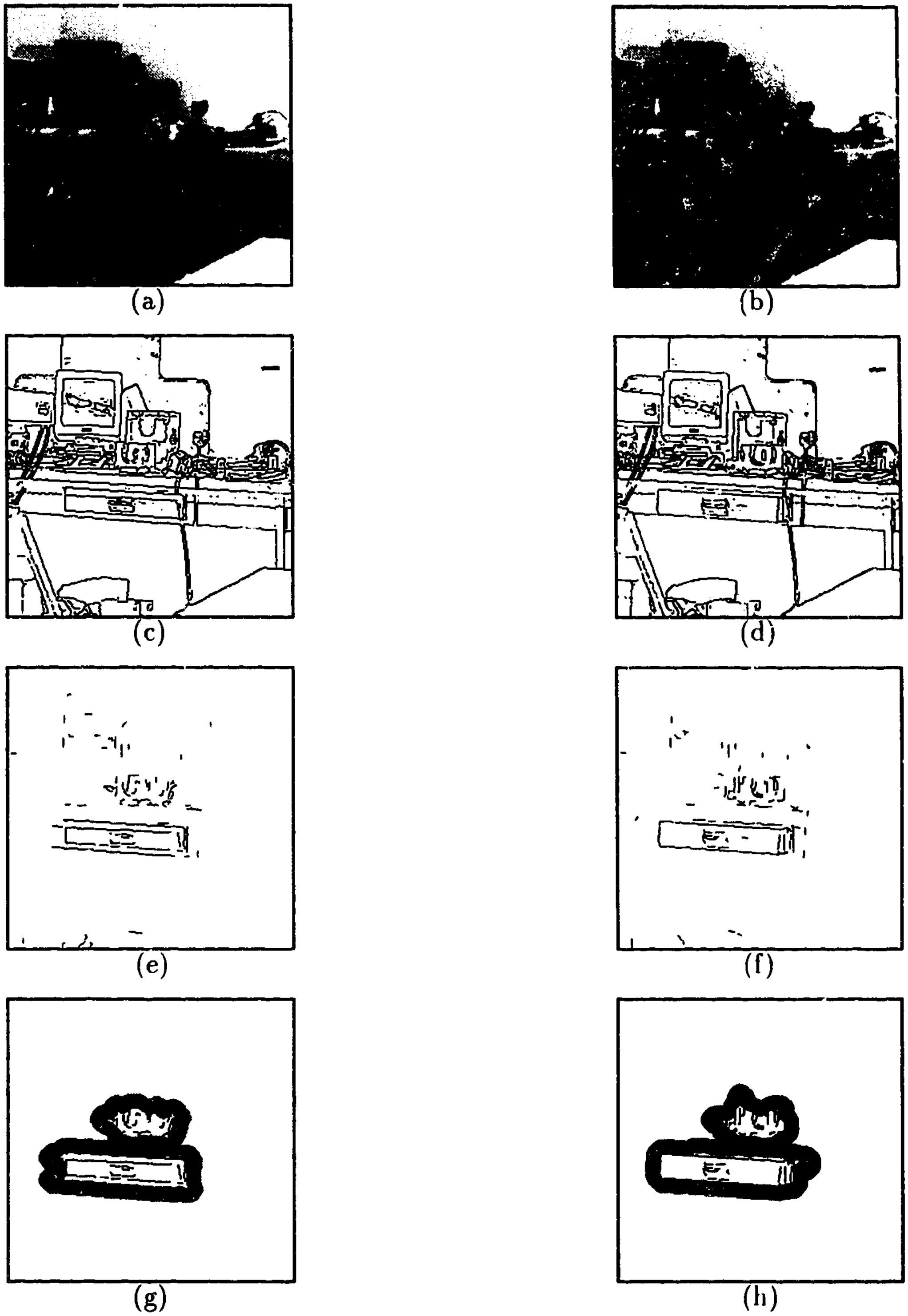


Figure 5.3: Results of applying the first moving object detection method to a pair of images

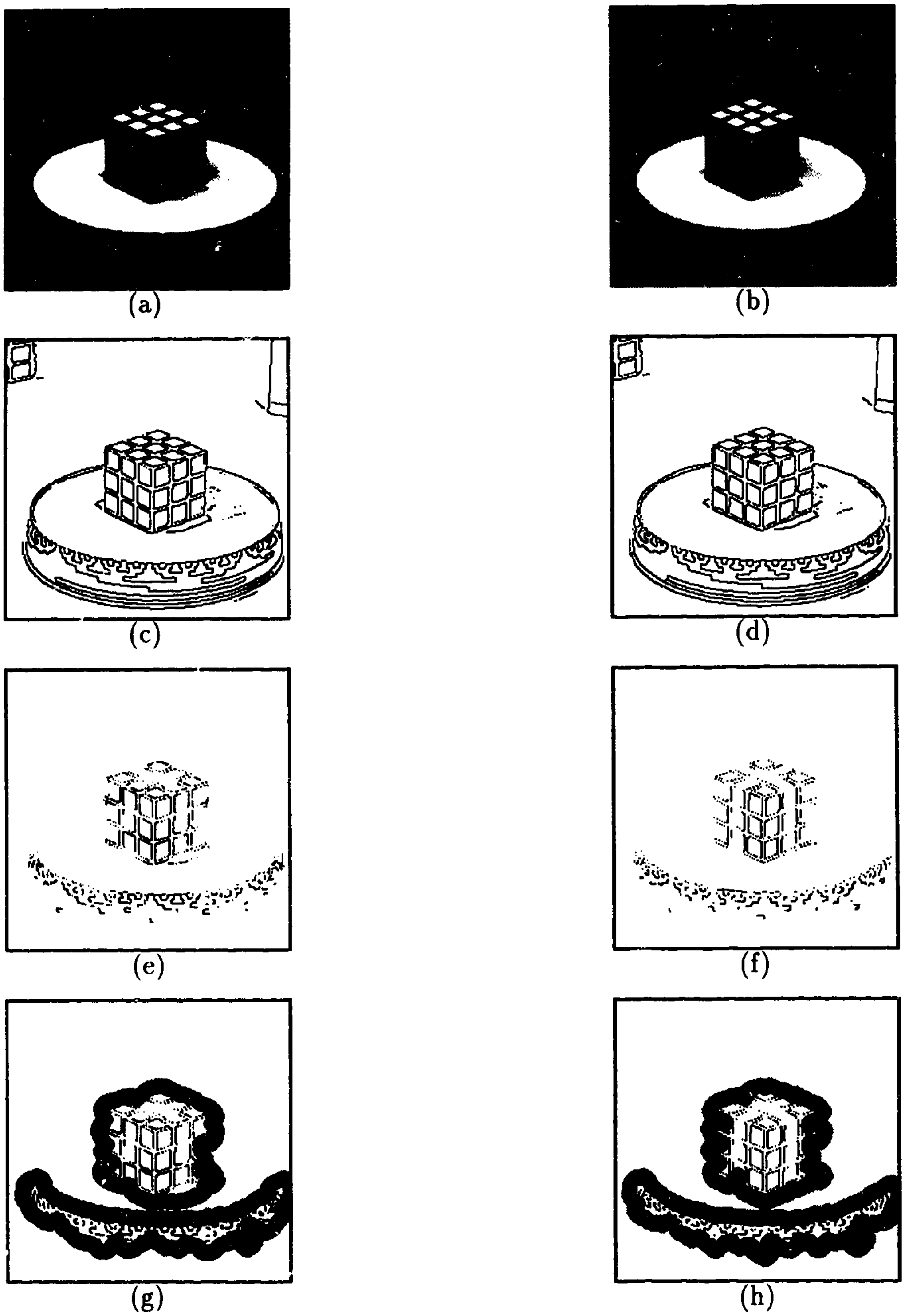


Figure 5.4: Results of applying the first moving object detection method to an image pair

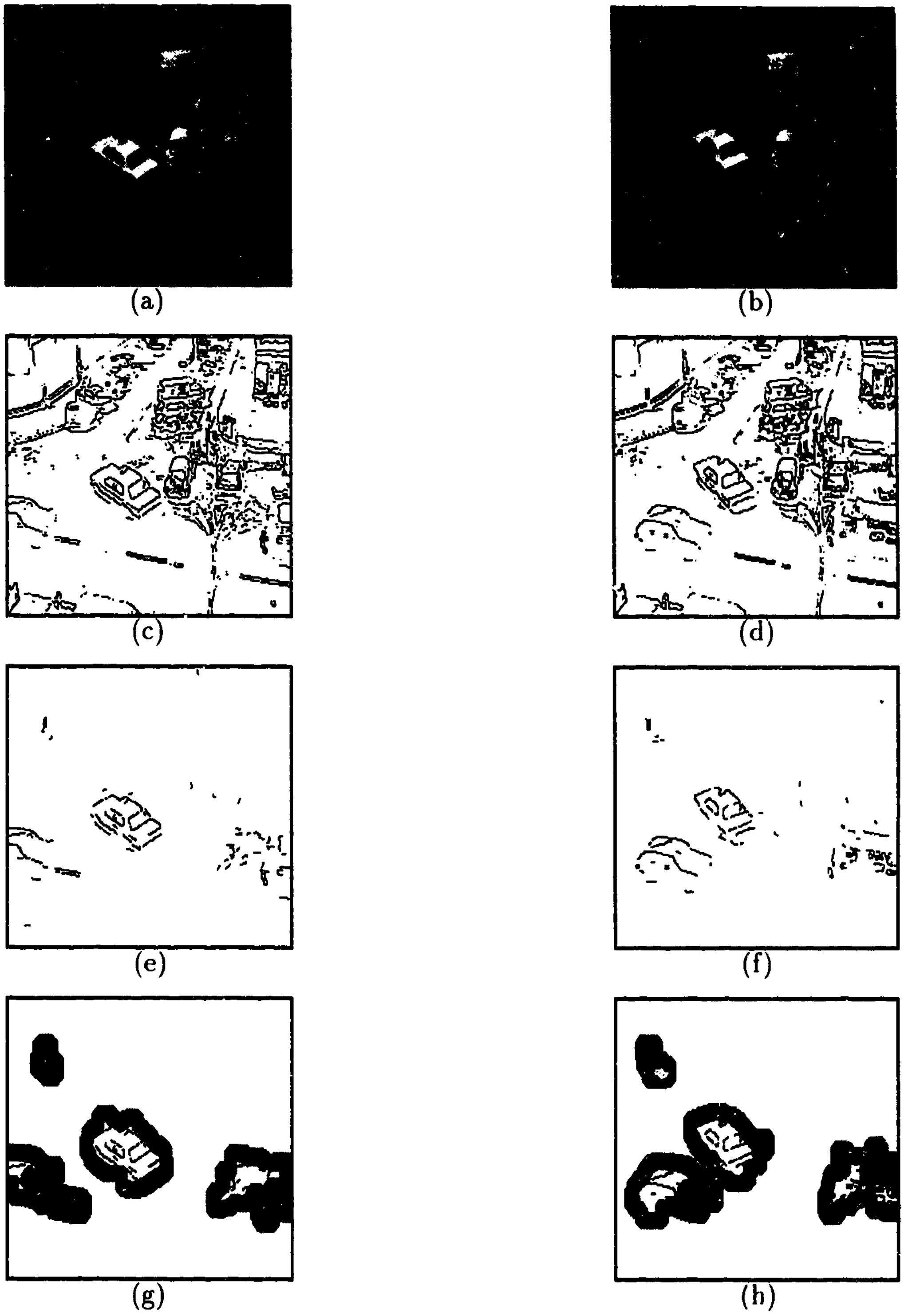


Figure 5.5: Results of applying the first moving object detection method to an image pair

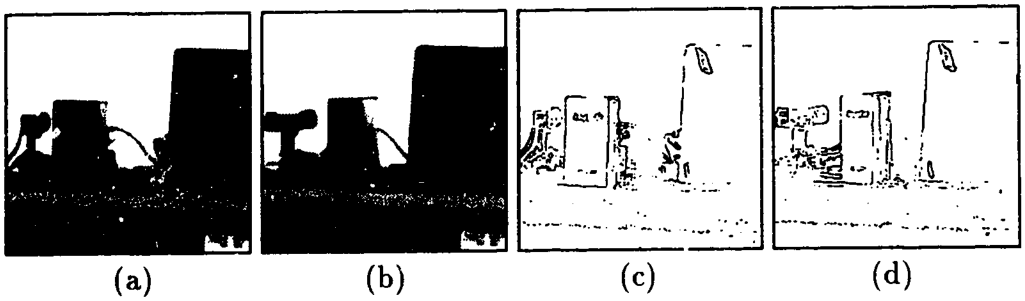


Figure 5.6: An example with overlapped boundaries

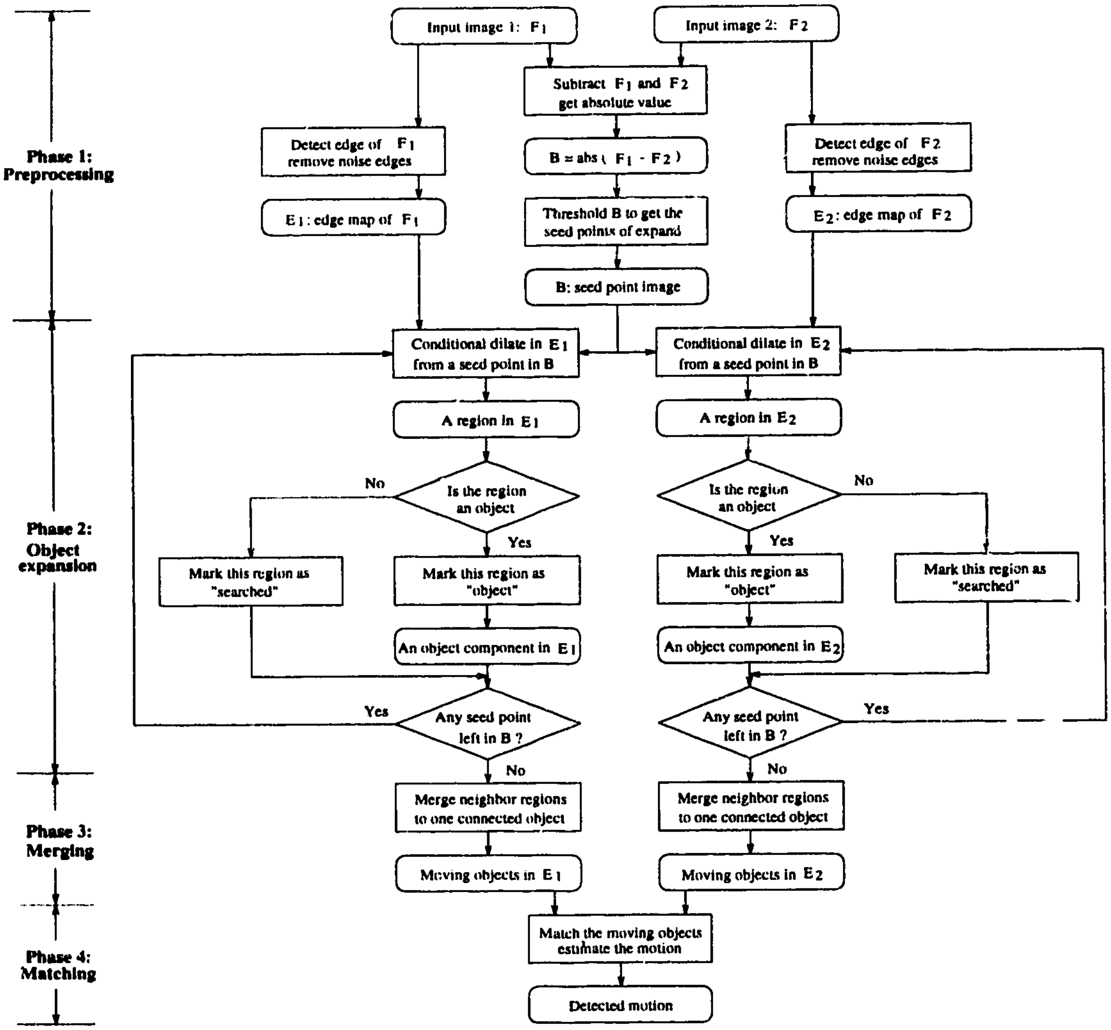

Phase 1: Preprocessing
Phase 2: Object expansion
Phase 3: Merging
Phase 4: Matching
Input image 1: F1
Input image 2: F2
Subtract F1 and F2 get absolute value
B = abs (F1 - F2)
Threshold B to get the seed points of expand
B: seed point image
Detect edge of F1 remove noise edges
E1 : edge map of F1
Detect edge of F2 remove noise edges
E2 : edge map of F2
Conditional dilate in E1 from a seed point in B
Conditional dilate in E2 from a seed point in B
A region in E1
A region in E2
Is the region an object
No
Yes
Mark this region as "searched"
Mark this region as "object"
An object component in E1
An object component in E2
Any seed point left in B ?
Yes
No
Merge neighbor regions to one connected object
Moving objects in E1
Moving objects in E2
Match the moving objects estimate the motion
Detected motion

Figure 5.7: Structure of the moving object detection algorithm based on expand dilation

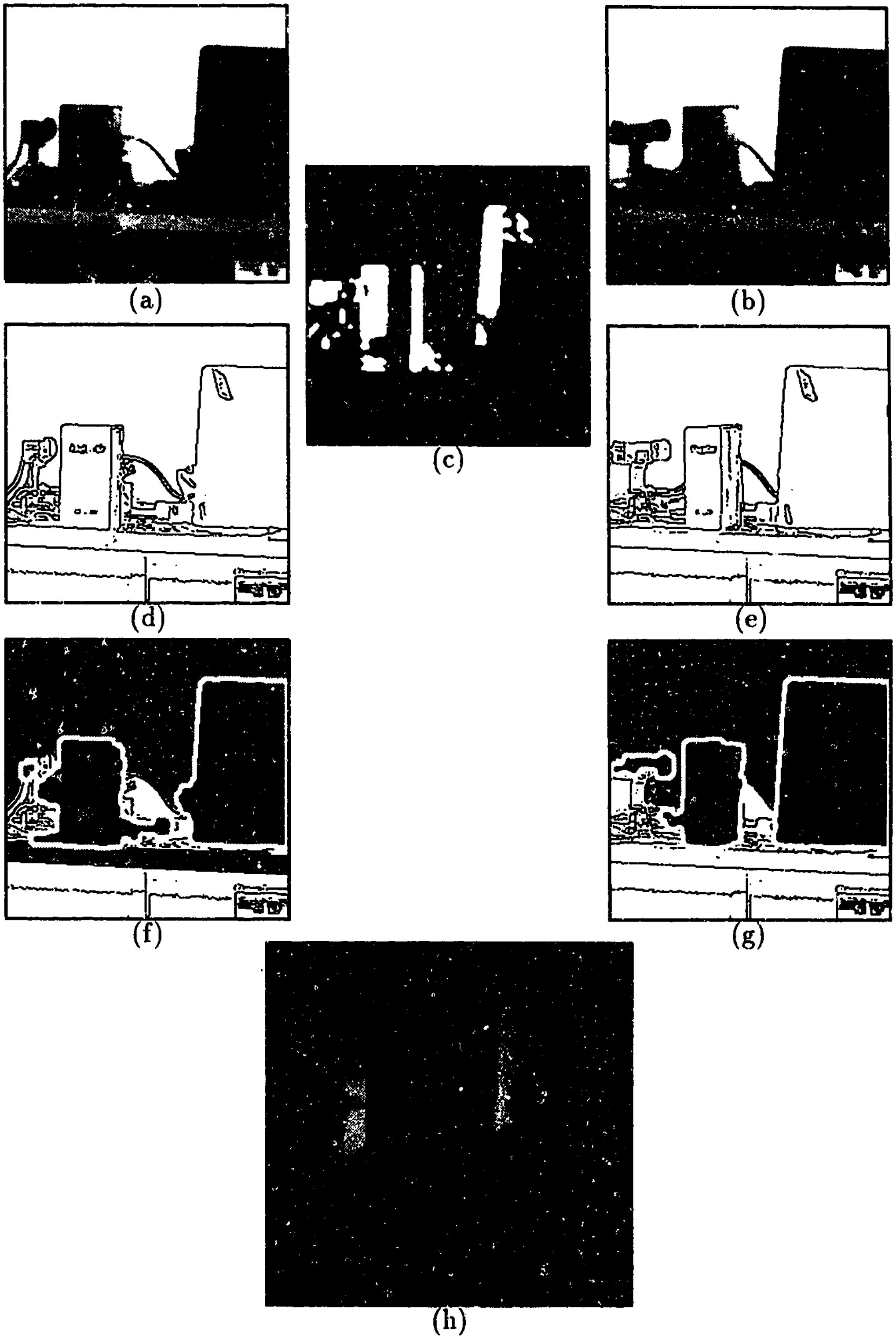


Figure 5.8: Results of applying the second moving object detection method to an image pair

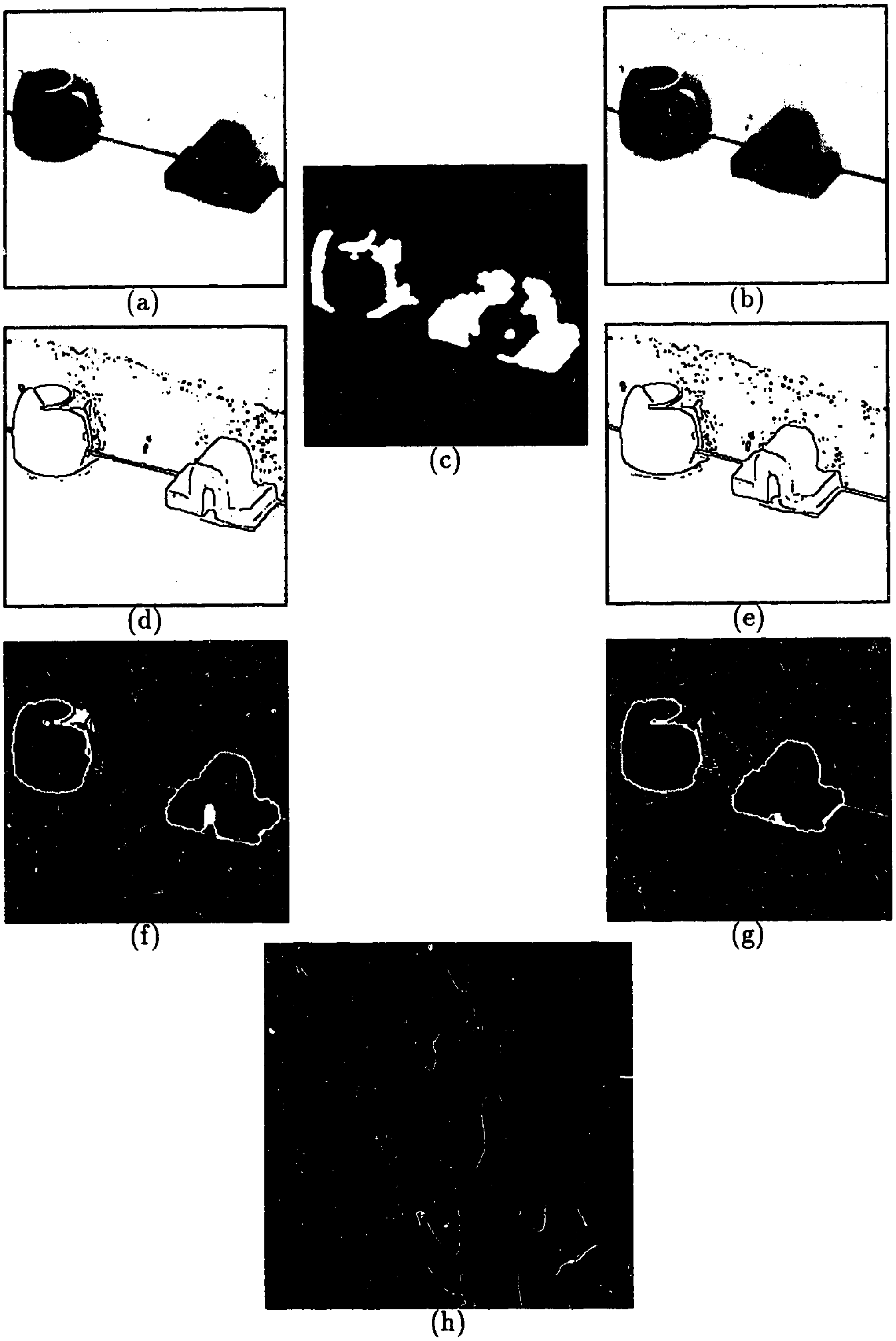


Figure 5.9: Results of applying the second moving object detection method to an image pair 2

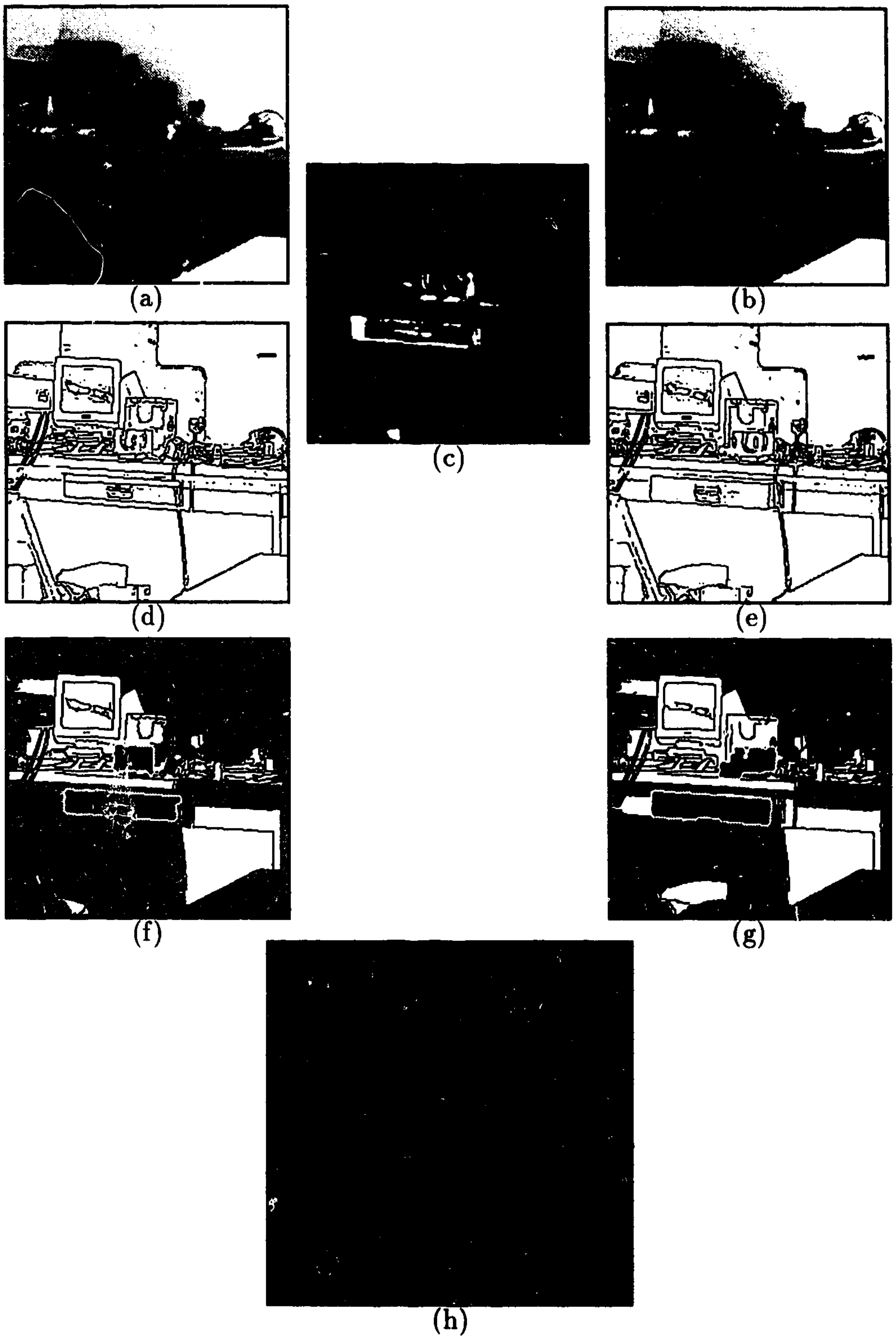


Figure 5.10: Results of applying the second moving object detection method to an image pair 3

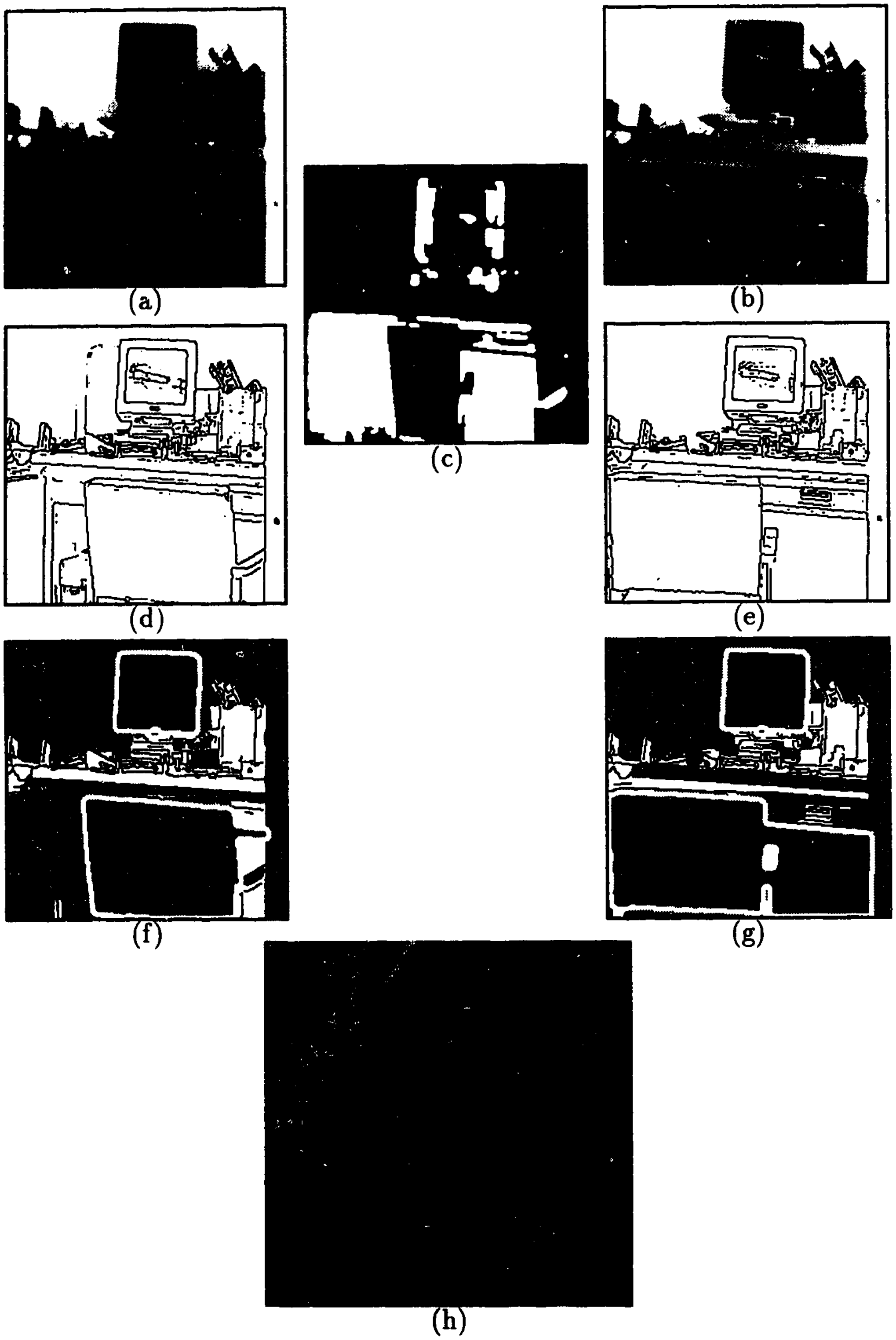


Figure 5.11: Results of applying the second moving object detection method to an image pair

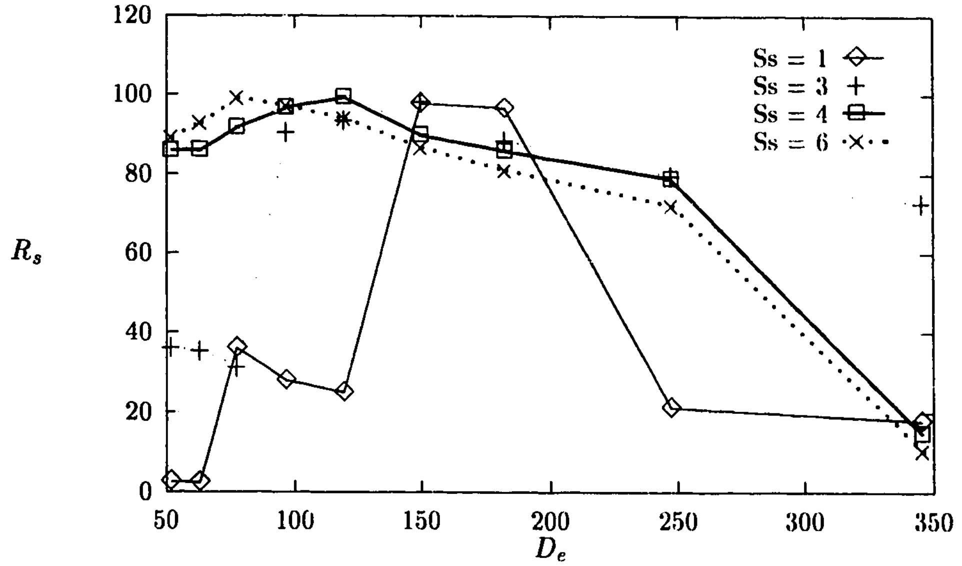


Figure 5.12: Relation between D_e and R_s, $T_d = 18$

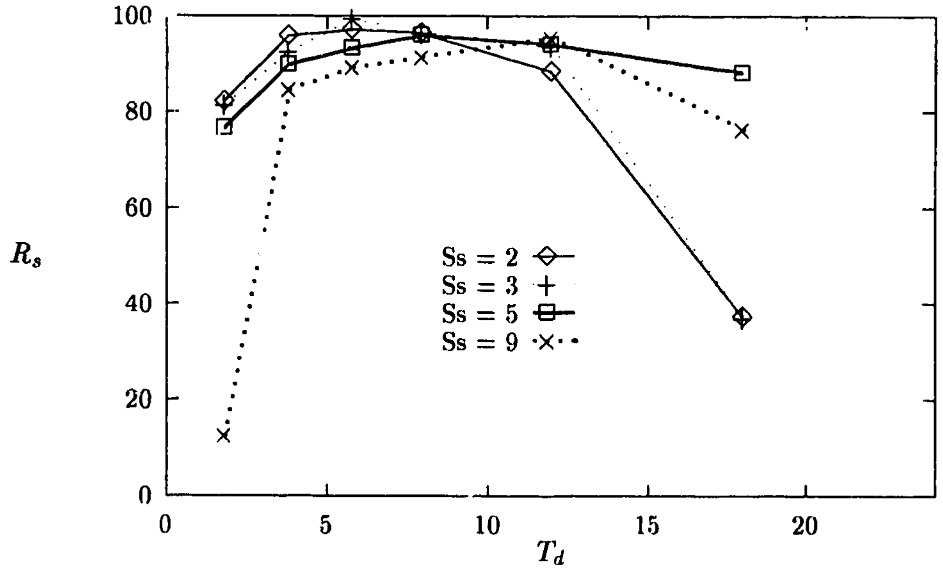


Figure 5.13: Relation between R_s and T_d

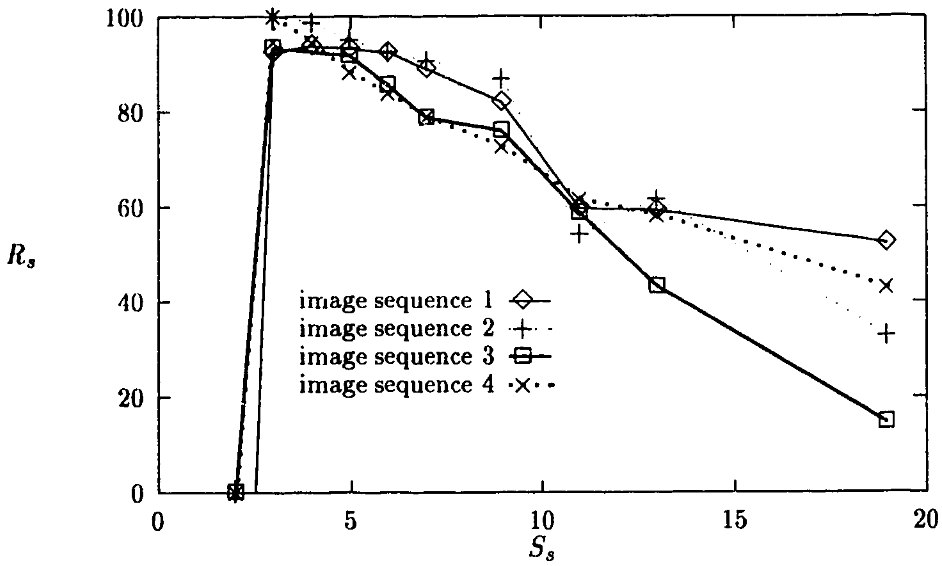


Figure 5.14: Relation between structuring element size and R_s

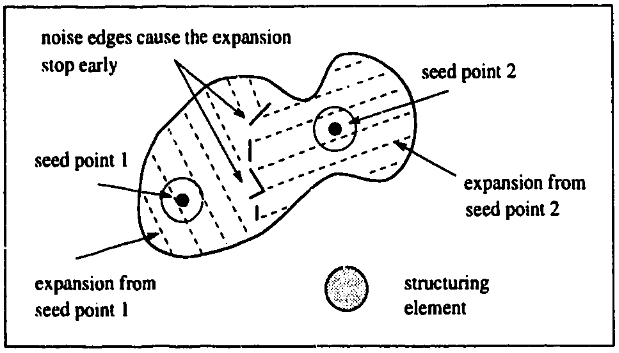

noise edges cause the expansion stop early
seed point 2
seed point 1
expansion from seed point 2
expansion from seed point 1
structuring element

Figure 5.15: An illustration of successful expansion with an additional seed point

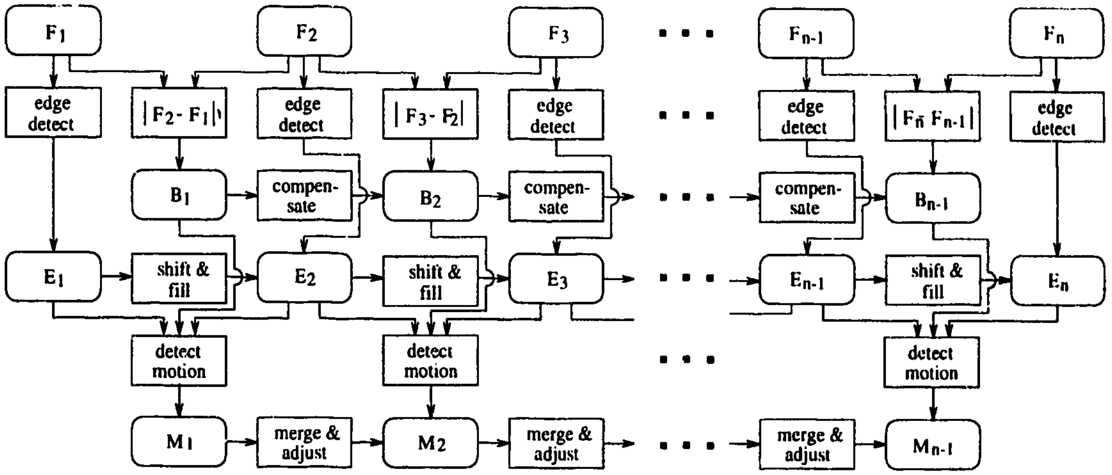


Figure 5.16: A Multi-frame moving object detection structure based on expand dilation

Chapter 6

Conclusions

In this thesis, a new concept, dynamic morphology is proposed. In traditional morphology, an operation is carried out on an image without preference to any pixel or any feature. Processing pixels of no interest is a unnecessary cost and may reduce the precision of the result. The difference between dynamic morphology and traditional morphology is that a dynamic morphological operation is an adaptive process which only deals with the pixels of interest. In this process, the move of the structuring element is dependent on the output of applying a specifically designed condition to certain pixels. A dynamic morphological operation cannot be replaced by a series of traditional morphological operations. Only operating on features of interest makes dynamic morphology more efficient and more precise. In this research two dynamic dilations, roll dilation and expand dilation are defined and discussed in detail. A boundary detection algorithm and two moving object detection algorithms are designed and implemented based on the dynamic dilations. The encouraging experimental results of applying these algorithms to different images demonstrated the capability and the potential of dynamic morphology.

The definitions in this thesis can be extended to other morphological operations and gray scale images or even multi-dimensional signals. In this work the conditions of dynamic dilations are based on edge images. For different applications, the condition might be defined on other image data such as gray value, histogram, FFT. It would be of great help if a general algorithm can be invented for defining conditions automatically on the information of an input image. It should also be an interesting work to explore the parallelism of dynamic morphology. For example in roll dilation, the next position selection process can be carried out in a parallel mode. A high [illegible] parallelism is that an object can be divided into several parts and roll dilation can be applied to each segment at the same time.

The structuring element plays an important role in a dynamic morphological operation. For unknown objects isotropic structuring elements can be used for a fair performance. If the structuring element shape matches the object shape, an optimal result can be expected. With prior knowledge of an object, structuring elements of specific shapes can be used for processing special image features. For example, a line shaped structuring element could be used to link broken straight lines. It would be worth while to research the properties of different structuring element in dynamic morphology and develop algorithms of choosing the best or the optimal structuring element for an certain image.

The proposed algorithms can also be improved to process images with much noise, more complicated objects and more complex backgrounds. The two moving object detection algorithms complement each other. They can be combined to form a more powerful system. A simple way is to build up a feedback frame, when one method failed the other one can be used. The best approach would be to study the original images and make the choice before hand. An adaptive system which makes use of some intermediate processing results is also a good idea.

Bibliography

[1] P. Anandan. A computational framework and an algorithm for the measurement of visual motion. *International Journal of Computer Vision*, 2:283–310, 1989.

[2] J. Bala and H. Wechsler. Shape analysis using morphological processing and genetic algorithms. In *Proceedings of Third International Conference on Tools for Artificial Intelligence*, pages 130–137, 1991.

[3] D. H. Ballard and C. M. Brown, editors. *Computer Vision.* Prentice-Hall Inc., 1982.

[4] H. Barman, L. Haglund, H. Knutsson, and G. Granlund. Estimation of velocity, accelaration and disparity in time sequences. In *Proceedings of Workshop on Computer Vision*, pages 44–51, 1991.

[5] J. L. Barron, D. J. Fleet, and S. S. Beauchemin. Systems and experiment performance of optical flow techniques. *Internationl Journal of Computer Vision*, 12(1):43–77, 1994.

[6] J. R. Bergen, P. J. Burt, R. Hinggorani, and S. Peleg. A three-frame algorithm for estimating two-component image motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(9):886–896, 1992.

[7] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Recognition and Machine Intelligence*, PAMI-8(6):679–698, 1986.

[8] J. Cooper, S. Venkatesh, and L. Kidtchen. Early jump-out corner detectors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(8):823-828, 1993.

[9] R. N. Dave. Boundary detection through fuzzy clustering. In *Proceedings of the 1992 IEEE International Conference on Fuzzy Systems*, pages 127–134, 1992.

[10] E. R. Dougherty, editor. *An introduction to morphological image processing.* SPIE Optical Engineering Press, 1992.

[11] M. P. Dubuisson and A. K. Jain. Object contour detection using color and motion. In *Proceedings of the 1993 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 471–476, 1993.

[12] J. H. Duncan and T. C. Chou. Temperal edges: The detection of motion and the computation of optical flow. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 374–382, 1988.

[13] A. Elnagar and A. Bsau. Robust detection of moving objects by a moving observer on planar surfaces. to appear.

[14] D. B. Goldgof, H. Lee, and T. Huang. Motion estimation without correspondence and object tracking over long time sequences. In *Proceedings of SPIE on Sense Fusion III: 3-D Perception and Recognition*, volume 1383, pages 109-121, 1990.

[15] R. M. Haralick, S. R. Sternberg, and X. Zhuang. Image analysis using mathematical morphology. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-9:532–549, 1987.

[16] B. K. P. Horn and B. G. Schunck. Determining optical flow. *Artificial Intelligence*, 17:185–204, 1981.

[17] D. C. D. Huang and K. T. Chen. Boundary detection based on neural network. In *Third International Conference on Tools for Artificial Intelligence*, pages 254–257, 1991.

[18] D. P. Huttenlocher, J. J. Noh, and W. J. Rucklidge. Tracking non-rigid objects in complex scenes. In *Proceedings of IEEE 4th International Conference on Computer Vision*, pages 93–101, 1993.

[19] M. K. Jain. Variable-resolution techniques for boundary detection and character thinning, 199[illegible]

[20] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. In *Proceedings of the First International Conference on Computer Vision*, pages 259–268, June 1987.

[21] V. Lacroix. A three-module strategy for edge detection. *IEEE Transactions on Pattern Analysis and Machine Vision*, 10(6):803–810, 1988.

[22] K. Lai and R. Chin. On regularization, formulation and initialization of the active contour models (snakes). In *Proceedings of Asian Conference on Computer Vision*, pages 542–545, 1993.

[23] K. Lai and R. Chin. Deformable contours: Modeling and extraction. In *Proceedings of the 1994 Computer Society Conference on Computer Vision*, pages 601–608, June 1994.

[24] X. Lee, P. Zhou, and A. Leon-Garcia. An efficient mpeg motion compensation scheme by motion trajectory tracking method. In *Proceedings of SPIE on Visual Communications and image Processing*, volume 1818, pages 594–605, 1992.

[25] S. Loncaric and A.P. Dhawan. A morphological signature transform for shape description. *Pattern Recognition*, 26(7):1029–1037, 1993.

[26] B. S. Manjunath and R. Chellappa. A computational approach to boundary detection. In *1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 358–363, 1991.

[27] G. Matheron, editor. *Random Sets and Integrai Geometry.* New York, Wiley, 1979.

[28] S. Menet, P. Saint-Marc, and G. Medioni. Active contour models: overview, implementation and applications. In *Proceedings of the IEEL international Conference on Systems, Man and Cybernetics*, pages 194–199, 1990.

[29] C. J. Moran. A morphological transformation for sharpening edges of feature before segmentation. *Computer Vision, Graphics, and Image Processing*, 49(1):85–94, 1990.

[30] D. Murray and A. Bsau. Motion tracking with an active camera. *IEEE Transactions on Pattern Analysis and Machine Vision*, 16(5):449–459, 1994.

[31] K. Naoao, M. Sohma, K. Kawakami, and S. Ando. Detecting contours in image sequences. *IEICE Transactions on Information and Systems*, E76-D(10):1162 1173, 1993.

[32] C. Nastar and N. Ayache. Fast segmentation, tracking and analysis of deformable objects. In *Proceedings of IEEE 4th International Conference on Computer Vision*, pages 275–279, 1993.

[33] D. Nichol and M. Fiebig. Tracking multiple moving objects by binary object forest segmentation. *Image and Vision Computing*, 9(6):362–371, 1991.

[34] C. M. Oranoe and F. C. A. Groen. Model based corner detection. In *Proceedings of the 1993 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 690–691, 1993.

[35] I. Pitas and A. N. Venetsanopoulos. Morphological shape decomposition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(1):38–45, 1990.

[36] I. Pitas and A. N. Venetsanopoulos. Morphological shape representation. In *Proceedings of the 1991 International Conference on Acoustics, Speech, and Signal Processing*, pages 2381–2384, 1991.

[37] A. Rattarabgsi and R. T. Chin. Scale-based detection of corners of planar curves. *IEEE Transactions on Pattern Recognition and Machine Intelligence*, 14(4):430–449, 1992.

[38] C. Ronse and B. Macq. Morphological shape and region description. *Signal processing*, 25(1):91–105, 1991.

[39] P. Saint-Marc, H. Rom, and G. Medioni. B-spline contour representation and symmetry detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(11):1191–1197, 1993.

[40] J. Serra, editor. *Image Analysis and Mathematical Morphology.* Academic Press, New York, 1982.

[41] F. Y. Shih and O. R. Mitchell. Skeletonization and distance transformation by greyscale morphology. In *Proceedings of SPIE Symposium on Automatic Inspection and high speed vision architectures*, 1987.

[42] D. Sinha and C. R. Giardina. Discrete black and white object recognition via morphological functions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(3):275–293, 1990.

[43] K. Sohn, W. E. Alexander, J. H. Kim, Y. Kim, and W. E. Snyder. Curvature estimation and unique corner point detection for boundary representation. In *Proceedings of the 1992 IEEE International Conference on Robotics and Automation*, pages 1590–1595, 1992.

[44] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: a fractorization method. *International Journal of Computer Vision*, 9(2):137-154, 1992.

[45] D. Williams and M. Shah. Edge contours using multiple scales. *Computer Vision, Graphics, and Image Processing*, 51:256–274, 1990.

[46] D. Williams and M. Shah. A fast algorithm for active contour and curvature estimation. *CVGIP: Image Understanding*, 55:14–26, 1992.

[47] Z. Zhang and O. D. Faugeras. Three-dimensional motion computation and object segmentation in a long sequence of stereo frames. *Internationl Journal of Computer Vision*, 7(3):211–241, 1992.

[48] D. Zhao and D. G. Daut. Shape recognition using morphological transformations. In *Proceedings of the 1991 International Conference on Acoustics, Speech, and Signal Processing*, pages 2565–2568, 1991.